Master's Thesis

Master's degree in Automatic Control and Robotics

Comparison of Active Learning Methods for Automatic Document Classification

THESIS

Author:	Marcé Gomis, Marc
University supervisor:	Ruiz Vegas, Francisco Javier
Company supervisor:	Michael, Jun Cai
Call:	20 June 2018

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Vilanova i la Geltrú, 20.06.2018

.....

(Signature [your name])

Abstract

Getting correctly labelled data is an important preliminary stage for many supervised machine learning problems, but it can also be really difficult to perform. Sometimes, to obtain a good model, it could require tens or hundreds of thousands of examples and examples usually does not come with labels. Active Learning is a methodology that can radically accelerate the labelling process and reduce costs for many machine learning projects. This methodology prioritises which data is most confusing and requests just those labels instead of collecting all the labels for all the data at once.

The aim of this work is to implement an Active Learning methodology that helps to automate the capture of relevant information from Key Investment Information Documents (KIID). Those documents aim to help investors to understand the nature and key risks of investment products in order to make a more informed investment decision.

Until now, data extraction was done through a specific labelling application developed by the company where this project was performed. This application is able to extract pieces of text from KIIDs, to assign its surrounding information and to query a certain human oracle (or expert) the correct label (whether the piece of text is associated or not to a specific type of information). The database obtained after this process is used in a supervised machine learning task to obtain a model that allows to recognise the type of information of a piece of text. The incorporation of the Active Learning methodology to this application will drastically reduce the query phase i.e. the most expensive part of the whole process.

The machine learning algorithm used for classification is the well-known Support Vector Machines. The examples are translated into binary vectors through Uni-grams and Bi-grams which means generating a vocabulary of *one* and *two* words and identifying whether the text feature contains these N-grams ($N = 1$ or 2) or not.

Three of the most common Active Learning strategies have been implemented in order to compare between them and with the passive learning method. Active learning strategies implemented are: *uncertainty sampling*, *query by committee* and *density-weighted* methods. All of them compared with the passive learning method, *random sampling*. 10-fold cross-validation is used to assess each of the approaches implemented and three evaluation measures are applied: recall, accuracy and computational time.

Contents

List of Figures

List of Tables

1 Introduction

Automating the key aspects of highly skilled knowledge work is going to become one of the most disruptive forces since the Industrial Revolution. This task is generally referred as Cognitive computing, that is, the task of building new hardware and/or software that mimics the functioning of the human brain and helps to improve human decision-making. Cognitive computing links Data analysis, that consists of inspecting, extracting, transforming and relating data with the goal of discovering useful information, suggesting conclusion and supporting decision-making.

Related terms such as Data mining, Big Data, Analytics, Cloud computing, Machine learning, Pattern recognition and Artificial Intelligence are invading more and more professional areas that have traditionally required human skills. Higher-skilled job categories in medicine, legal services, accounting, finance and law enforcement are all in scope to be replaced, to a large degree, by cognitive technologies.

Most of machine learning algorithms are huge *guess-and-check* machines. They take some data, calculate a guess, check their answer, adjust a little bit and try again with some new data. Over lots of data, the algorithm can become very accurate. A critical part of this process is having the *"right"* answers available for the algorithm to check against, *its labels*. Labels depend on the problem. If the problem is a spam detection, for instance, the labels will be *"Spam"* or *"Not Spam"*. Whereas if the problem is to determine the text mood, the labels will be *"happy", "annoyed"* or *"sad"*.

Getting correct labelled data is important, but it can also be really hard to acquire. Sometimes, to obtain a good model it could require tens or hundreds of thousands of examples and examples usually does not come with labels. So, "experts" have to review the data and provide the "right" labels. In most cases anyone could be an "expert" and can label data, for example for a spam detection problem. Eventually the problem requires very skilled technicians like in the case of cancer cell detection. Getting enough expertise to label enough data can be very expensive.

"Getting labelled data is a huge and very often a prohibitive cost for a lot of machine learning projects."

Active Learning (AL) is a methodology that can radically accelerate the process and reduce costs for many machine learning projects. It can sometimes largely reduce the amount of labelled data required by prioritising the labelling work for the experts. AL prioritises which data is most confused about and requests just those labels instead of

collecting all the labels for all the data at once. Then the algorithm trains with the new reduced set of labelled data and repeats asking for some more labels among the most confusing data.

Using this methodology, experts can focus on labelling the most informative data, providing the most useful information with the least amount of time. This helps the algorithm to learn faster and lets the experts skip labelling data that would not be very helpful to the model. Consequently time and money are saved.

1.1 Motivation

Nowadays there are a lot of very accessible sources of information for investment products: in print, radio and television, in the Internet, in bank offices and several other places. Nowadays this information can be found in more or less standarized documents called a Key Investment Information Document (KIID). Those documents are a few pages (normally 2) which include the investment product critical information. KIIDs aims to help investors understand the nature and key risks of the product in order to make a more informed investment decision.

An investment fund is a supply of capital belonging to numerous investors used to collectively purchase securities while each investor retains ownership and control of his own shares. With investment funds individual investors do not make decisions about how assets of a fund should be invested. They simply choose a fund based on its goals, the risk, fees and other factors. In other words, an investment fund is a way of investing money alongside other investors in order to benefit from the inherent advantages of working as part of a group.

With thousands of funds to choose from, investors and their advisers require factual information to make more informed choices. Consequently, recent regulation has led to the introduction of the Key Investor Information Document (KIID) which aims to provide investors with a transparent and succinct overview of funds in a common format, before they invest.

The KIID follows a standard format which comprises several sections, such as risks and past performance. KIIDs provide investors with important information on the fund to help determine if it aligns with their investment goals, time horizon and risk tolerance. They can be found all over the network in web pages from investment companies or directly requested in investment companies. Although finding a KIID is an easy task, investors need to analyse those documents in order to hit the target.

To facilitate this task, some comparison tools currently exist in the market. For instance *Fidelity international*[1] provides a comparison interface where the user can

compare several funds by selecting a few variables such as fund provider and on-going charge. *Moneywise*[2] is a platform where different funds can be compared based on their past performance even though past performance is not a reliable indicator to future returns. *Vanguard*[3] is another tool in which the funds can be compared to whether the investor previously knew the name of the fund. All these tools manually update its funds information monthly, weekly or even daily. This is a costly task because of the number of funds that currently are in the market. Consequently, those tools only provide few information of KIIDs.

This work focuses on the automation of extracting information from KIIDs. The current process need to download, read and interpret each investment document and update the information in the comparison tool as *Moneywise* does. For that reason, there exists the necessity of automating the data extraction in investment documents. In this project different AL techniques are proposed in order to automate the information extraction process diminishing the enterprise cost.

1.2 Objectives

The aim of this work is to implement a methodology that automates the information extraction from KIIDs minimising the labelling cost[4]. It is intended to extract automatically important information for investment taking decisions. Information which is extracted from KIIDs is:

1. **Exit charge**: Fee charged to investors when they redeem shares from a fund.
2. **Fund launch date**: Date on which the fund began its operations.
3. **Available languages**: Languages in which the document, (KIID), is available.

This information is extracted using a machine learning method and employing different AL techniques.

In order to accomplish these objectives the following sub-goals were set up:

1. Study the state of the art for AL methods and feature extraction techniques for automatic document classification.
2. Generate a database formed by text from KIIDs in order to develop an automatic extraction and classification algorithm that allows to identify information previously presented. A database is formed by pieces of text extracted from those investment documents together with its context. In order to obtain such database the following steps are completed:
   a) Obtaining investment documents from fund investment companies.
   b) Labelling the required information in the documents.

3. Preprocessing the text data in order to apply the chosen algorithm.

4. Applying the text classification algorithms over the database.

5. Analysing the results and extract conclusions. AL strategies will be revised in order to implement the best one in a real time annotation interface.

For example, suppose that an investor asks the company to know which funds launched after 2016 **(fund launch date)** contain the highest exit charges **(exit charge)** and are available in English and French **(available languages)**. In this manner the company needs to find all the KIIDs which funds where launched after 2016, have the highest exit charges and are available in English and French in the Internet, read them, obtain the ordered information and write up a summary for the costumer. The system that is developed in this project will allow to automatically request the investment documents and extract the required information.

1.3 Planning

This project was developed in a 4 month duration. In order to achieve the objectives presented in the previous section, the following planning was made.

1. Study the state of the art: the first part of this project is the research and learning of the work. This project aims to meet the needs which the current market technology is not able to cover yet. Therefore, it was needed to study about text feature extraction, classification methods for text classification and AL methods.

2. Generate the database: it was necessary to make a text database with piece of text extracted from KIIDs jointly with its contextual information derived from different investment information sources.

3. Study the main AL query methods: in the state of the art, various AL strategies used in previous works were identified. In this task the most promising methods were studied in depth in order to be implemented.

4. Preprocess the data: in order to use the data extracted from KIIDs, they have to be preprocessed to obtain the text features and to condition the data to be used in the classifiers.

5. Compare the algorithms: in this section the algorithms were trained using a cross-validation technique for assessing how the results generalise over an independent data set.

6. Analyse and discuss the results.

7. Write up the report: report writing was done during all the project development.

2 State of the art

This section is intended to give an introduction about relevant terms, techniques and standards in the field of AL.

Machine Learning (ML) plays a key role in a wide range of applications, such as data mining, natural language processing and image recognition. ML is an application of Artificial Intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Learning process begins with data observations, such as examples, direct experience, or instruction. The objective is to look for data patterns in order to make better future decisions based on the provided examples. The primary aim is to allow computers to learn automatically without human intervention or assistance and to adjust actions accordingly.

ML algorithms are often categorized as supervised and unsupervised.

- Supervised ML algorithms can take decisions based on what has been learned in the past using labelled examples. Starting from the analysis of a known training data set, the learning algorithm produces an inferred function to make predictions. The system is able to provide targets for any new input after enough training. The learning algorithm can also compare its output with the correct ones and find errors in order to validate the model.
- In contrast, unsupervised ML algorithms are used when the information used is not labelled. Unsupervised learning studies how systems can discover a hidden structure from unlabelled data.

2.1 Active Learning

AL, also called *query learning*, is a branch of ML where the learning algorithm is allowed to query the training data set. The key hypothesis is that if the learning algorithm is allowed to choose the data from which it learns, it will perform better with less training. This is a desirable property for any learning algorithm.

For any supervised learning system to perform well, it must often be trained on hundreds or even thousands of labelled instances. Sometimes these labels come at little or no cost, such as the "spam" flag marked on unwanted email messages, or the five-star

film rating on a social networking website. Learning systems use these flags and ratings to better filter junk email and suggest movies. In these cases such labels are provided for free, but for many other more sophisticated supervised learning tasks, labelled instances are very difficult, time-consuming, expensive to obtain or even costly e.g. speech recognition, information extraction and document classification and filtering.

2.1.1 Scenarios

There are several scenarios in which the active learner may pose queries, and there are also several different query strategies that were used to decide which instances are most informative. The three main scenarios are:

- **Membership Query Synthesis**. In this scenario, the learner constructs examples for labelling for any unlabelled instance in the input space, rather than those sampled from some underlying natural distribution. Query synthesis is reasonable for many problems, but labelling such arbitrary instances can be awkward if the oracle is a human annotator. For example, [5] employed membership query learning with human oracles to train a neural network to classify handwritten characters. They encountered that many of the query images generated by the learner contained symbols without natural semantic meaning.

- **Stream-based selective sampling**. An alternative to synthesising queries is selective sampling. It consists in sampling the unlabelled instance from the actual distribution, and then the learner can decide whether or not to request its label. Each unlabelled instance is typically selected one at a time from the data source. It is a suitable scenario if obtaining an unlabelled instance is free or in-expensive. The decision whether or not to query an instance is normally taken by means of an information rating measure. Instances of which the measure value is above some threshold are then queried.

Another important approach is to define the region that is still unknown to the overall model class, i.e. to the set of hypotheses consistent with the current labelled training set called *version space*[6].

- **Pool-based sampling**. This scenario is related to problems where large collections of unlabelled data can be gathered at once[7]. Here, it is assumed that there is a small set of labelled data and a large static or non-changing pool of unlabelled data. The pool-based scenario has been studied for many real-world problem domains in ML, such as text classification, information extraction, image classification and retrieval, video classification and retrieval, speech recognition, cancer diagnosis, etc.

The main difference between stream-based and pool-based AL is that the former scans through the data sequentially and makes query decisions individually, whereas the later evaluates and ranks the entire collection before selecting the best query. While the pool-based scenario appears to be much more common among application papers, one can

imagine settings where the stream-based approach is more appropriate. For example, when memory or processing power may be limited, as with mobile and embedded devices.

2.2 Query Strategy Frameworks

All AL scenarios involve evaluating the information rating of unlabelled instances, which can either be generated de novo or sampled from a given distribution. There have been many proposed ways of formulating such query strategies in the literature.

2.2.1 Uncertainty sampling

Perhaps the simplest and most commonly used query framework is uncertainty sampling. In this framework, an active learner queries the instances about which it is least certain how to label. This approach is straightforward for probabilistic learning models. For instance, when using a probabilistic model for binary classification, uncertainty sampling simply queries the instance at which posterior probability of being positive is nearest 0.5[7]. For problems with more than two class labels several approaches exists:

- **Least confident:** Query the instance whose prediction is the least confident or the class label with the highest posterior probability under the model θ.

$$x^*_{LC} = argmax_x(1 - P_\theta(\hat{y}|x)), \text{ where } \hat{y} = argmax_y(P_\theta(y|x))$$

Example: In table 2.1 there is depicted the probability of 8 samples $(x_1, x_2, ..., x_8)$ belonging to 3 different classes (y_1, y_2 and y_3).

Sample	y_1	y_2	y_3
x_1	0.1	0.8	0.1
x_2	0.3	0.1	0.6
x_3	0.1	0.1	0.8
x_4	0.1	0.0	0.9
x_5	0.1	0.6	0.3
x_6	0.3	0.2	0.5
x_7	0.8	0.2	0.0
x_8	0.6	0.3	0.1

Table 2.1: Example 1 and 2 class probabilities.

Considering the least confident approach the prediction calculated looking for the sample with the highest posterior probability under the model. In this example it is

found that the sample with the highest posterior probability is x_6.

$$\begin{array}{ll}
\hat{y_1} = argmax_y(P_\theta(y|x_1)) = y_2 = 0.8 & 1 - P_\theta(\hat{y_1}|x_1) = 0.2 \\
\hat{y_2} = argmax_y(P_\theta(y|x_2)) = y_3 = 0.6 & 1 - P_\theta(\hat{y_2}|x_2) = 0.4 \\
\hat{y_3} = argmax_y(P_\theta(y|x_3)) = y_3 = 0.8 & 1 - P_\theta(\hat{y_3}|x_3) = 0.2 \\
\hat{y_4} = argmax_y(P_\theta(y|x_4)) = y_3 = 0.9 & 1 - P_\theta(\hat{y_4}|x_4) = 0.1 \\
\hat{y_5} = argmax_y(P_\theta(y|x_5)) = y_2 = 0.6 & 1 - P_\theta(\hat{y_5}|x_5) = 0.4 \\
\hat{y_6} = argmax_y(P_\theta(y|x_6)) = y_3 = 0.5 & 1 - P_\theta(\hat{y_6}|x_6) = \mathbf{0.5} \\
\hat{y_7} = argmax_y(P_\theta(y|x_7)) = y_1 = 0.8 & 1 - P_\theta(\hat{y_7}|x_7) = 0.2 \\
\hat{y_8} = argmax_y(P_\theta(y|x_8)) = y_1 = 0.6 & 1 - P_\theta(\hat{y_8}|x_8) = 0.4
\end{array}$$

$$x^*_{LC} = x_6$$

The criterion for the least confident approach only considers information about the most probable label without taken into account information about the remaining label distribution.

- **Margin sampling:** In order to consider the information about the remaining label distribution, some researchers use a different multi-class uncertainty sampling variant called *margin sampling*[8].

$$x^*_M = argmin_x(P_\theta(\hat{y_1}|x) - P_\theta(\hat{y_2}|x)),$$

where $\hat{y_1}$ and $\hat{y_2}$ are the first and second most probable class labels under the model θ, respectively.

Example: Considering the margin sampling approach the prediction calculated considering the first and the second most probable class labels using the class probabilities from table 2.1. In this example we found that the samples with the lowest probability difference is x_6.

$$\begin{array}{l}
P_\theta(\hat{y_1}|x_1) - P_\theta(\hat{y_2}|x_1) = 0.7 \\
P_\theta(\hat{y_1}|x_2) - P_\theta(\hat{y_2}|x_2) = 0.3 \\
P_\theta(\hat{y_1}|x_3) - P_\theta(\hat{y_2}|x_3) = 0.7 \\
P_\theta(\hat{y_1}|x_4) - P_\theta(\hat{y_2}|x_4) = 0.8 \\
P_\theta(\hat{y_1}|x_5) - P_\theta(\hat{y_2}|x_5) = 0.3 \\
P_\theta(\hat{y_1}|x_6) - P_\theta(\hat{y_2}|x_6) = \mathbf{0.2} \\
P_\theta(\hat{y_1}|x_7) - P_\theta(\hat{y_2}|x_7) = 0.6 \\
P_\theta(\hat{y_1}|x_8) - P_\theta(\hat{y_2}|x_8) = 0.3
\end{array}$$

$$x^*_M = x_6$$

Margin sampling aims to correct for a shortcoming in least confident strategy, by incorporating the posterior of the second most likely label.

- **Entropy:** Possibly the most popular uncertainty sampling strategy uses entropy[9] as an uncertainty measure:

$$x_H^* = argmax_x - \sum_i P_\theta(y_i|x) log P_\theta(y_i|x),$$

where y_i ranges all possible labels. Entropy is an information-theoretic measure that represents the amount of information needed to "encode" a distribution.

Example: In table 2.2 there is depicted the probability of 8 samples $(x_1, x_2, ..., x_8)$ belonging to 3 different classes (y_1, y_2 and y_3).

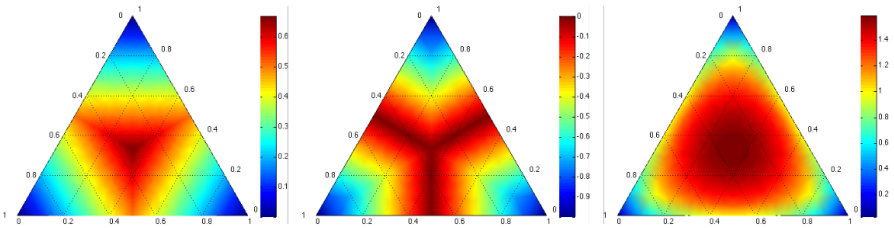
Sample	y_1	y_2	y_3	(1)	(2)	(3)	(4)
x_1	0.1	0.8	0.1	-0.10	-0.08	-0.10	0.28
x_2	0.3	0.1	0.6	-0.16	-0.10	-0.13	0.39
x_3	0.1	0.1	0.8	-0.10	-0.10	-0.08	0.28
x_4	0.1	0.0	0.9	-0.10	-0.00	-0.04	0.14
x_5	0.1	0.6	0.3	-0.10	-0.13	-0.16	0.39
x_6	0.3	0.2	0.5	-0.16	-0.14	-0.15	**0.45**
x_7	0.8	0.2	0.0	-0.08	-0.14	-0.00	0.22
x_8	0.6	0.3	0.1	-0.13	-0.16	-0.10	0.39

Table 2.2: Entropy probabilities.

$(1) = P_\theta(y_1|x) log P_\theta(y_1|x)$
$(2) = P_\theta(y_2|x) log P_\theta(y_2|x)$
$(3) = P_\theta(y_3|x) log P_\theta(y_3|x)$
$(4) = -\sum_i P_\theta(y_i|x) log P_\theta(y_i|x)$

For binary classification, entropy-based sampling reduces to the margin and least confident strategies above. In fact all three approaches are equivalent to querying the instance with a class posterior closest to 0.5. In this example we found that the samples with the highest probability is x_6.

Figure 2.1 shows the implicit relationship among these uncertainty measures. In all cases, the most informative instance would lie at the centre of the triangle, because this represents where the posterior label distribution is most uniform. Similar, the least informative instances are at the three corners, where one of the classes has a extremely



(a) least confident (b) margin (c) entropy

Figure 2.1: Uncertainty query behaviour measures in a three-label classification problem.

high probability.

The main differences are found in the rest of the probability space. For example, the entropy measure does not favour instances where only one of the labels is highly unlikely as along the outer side edges. The model is therefore fairly certain that it is not the true label. The least confident and margin measures, on the other hand consider such instances to be useful if the model cannot distinguish between the remaining two classes.

2.2.2 Query-By-Committee

Another, more theoretically-motivated query selection framework is the query-by-committee (QBC) algorithm[10]. The QBC approach involves maintaining a committee $C = \theta^{(1)}, ..., \theta^{(C)}$ of models which are all trained on the current labelled set $\mathcal{L}$ however represent competing hypotheses. The most informative query is considered to be the instance at which models most disagree. In a simple QBC version just two members are selected, two random hypothesis which belong to the same version space. A sample is labelled only if both hypothesis disagree on the predicted label. With this QBC version, several theoretical results have been achieved[10].

The fundamental premise behind the QBC framework is minimising the version space. QBC aims to constrain as much as possible the version space by querying in controversial regions of the input space. Figure 2.2 illustrates the concept of version spaces for (a) linear functions and (b) axis-parallel box classifiers in different binary classification tasks.

QBC algorithm requirements:

- Construct a committee of models representing different regions of the version space.
- Measure the disagreement among committee members.

There are several level of disagreement measures, one of them is *vote entropy*[11]:

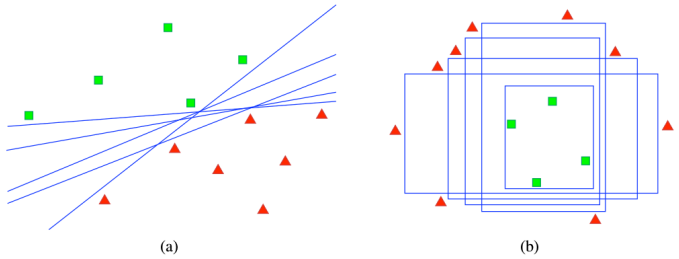


Figure 2.2: Version space examples for (a) linear and (b) axis-parallel box classifiers.

$$x^*_{VE} = argmax_x - \sum_i \frac{V(y_i)}{C} log \frac{V(y_i)}{C},$$

where y_i again ranges over all possible labels, and $V(y_i)$ is the number of "votes" that a label receives from among the committee members' predictions, and C is the committee size.

Example: In table 2.3 there is depicted the votes that a label receives from among the committee members' predictions of 8 samples $(x_1, x_2, ..., x_8)$. Committee size is $C = 5$.

Sample	$V(y_1)$	$V(y_2)$	$V(y_3)$	(1)	(2)	(3)	(4)
x_1	3	1	1	-0.13	-0.14	-0.14	0.41
x_2	1	3	1	-0.14	-0.13	-0.14	0.41
x_3	0	5	0	-0.00	0.00	-0.00	0.00
x_4	2	2	1	-0.16	-0.16	-0.14	**0.46**
x_5	0	5	0	-0.00	-0.00	-0.00	0.00
x_6	0	0	5	-0.00	-0.00	-0.00	0.00
x_7	1	1	3	-0.14	-0.14	0.13	0.41
x_8	3	1	1	-0.16	-0.14	0.16	0.29

Table 2.3: QBC example.

$(1) = \frac{V(y_1)}{C} log \frac{V(y_1)}{C}$, $(2) = \frac{V(y_2)}{C} log \frac{V(y_2)}{C}$, $(3) = \frac{V(y_3)}{C} log \frac{V(y_3)}{C}$, $(4) = -\sum_i \frac{V(y_i)}{C} log \frac{V(y_i)}{C}$

In this example it is found that the samples with the highest committee disagreement is x_4.

2.2.3 Expected Model Change

Another general AL framework uses a decision-theoretic approach and selecting the instance that would impart the greatest change to the current model if we knew its label. An example query strategy in this framework is the "expected gradient length" (EGL) approach for discriminative probabilistic model classes. In other words, the learner should query the instance x which, if labelled and added to $\mathcal{L}$, would result in the new training gradient of the largest magnitude.

$$x^*_{EGL} = argmax_x \sum_i P_\theta(y_i|x) \left\| \nabla l_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \right\|,$$

where $\|\cdot\|$ is, in this case, the Euclidean norm and $\nabla_\theta(L \cup \langle x, y_i \rangle)$ is the gradient of the objective function l with respect to the model parameters θ taken into account the training set L and the new pattern $\langle x, y_i \rangle$.

2.2.4 Expected Error Reduction

Another decision-theoretic approach that aims to measure how much its generalisation error is likely to be reduced. The idea is to estimate the expected future error of a model trained using $\mathcal{L} \cup \langle x, y \rangle$ on the remaining unlabelled instances in $\mathcal{U}$, and query the instance with minimal expected future error, the minimal risk. The objective here is to reduce the expected total number of incorrect predictions. One approach is to minimise the expected 0/1-loss:

$$x^*_{0/1} = argmin_x \sum_i P_\theta(y_i|x) \left(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x, y_i \rangle}}(\hat{y}|x^{(u)}) \right),$$

where $\theta^{+\langle x, y_i \rangle}$ refers to the the new model after it has been re-trained with the training tuple $\langle x, y_i \rangle$ added to $\mathcal{L}$. The objective in this approach is to reduce the expected total number of incorrect predictions.

2.2.5 Density-Weighted Methods

The main idea of the Density-Weighted Methods is that informative instances should not only be those which are uncertain, but also those which are "representative" of the underlying distribution. Therefore, queries are instantiated as follows:

$$x^*_{ID} = argmax_x \phi_A(x) \cdot \left(\frac{1}{U} \sum_{u=1}^{U} sim(x, x^{(u)}) \right)^{\beta},$$

where $\phi_A(x)$ represents the information rating of x according to some "base" query strategy A, such as an Uncertainty Sampling or QBC approach. The second term weights the information rating of x by its average similarity to all other instances in the input distribution, subject to a parameter β that controls the relative importance of the density term.

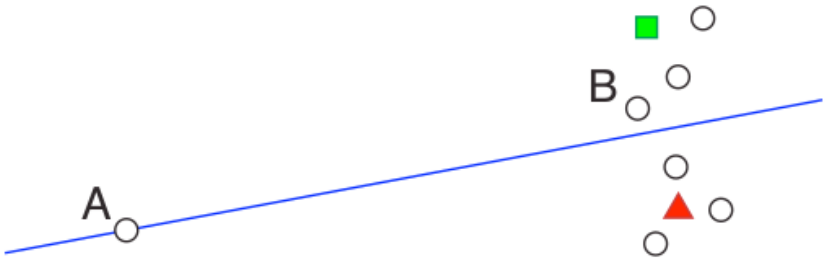


Figure 2.3: Illustration of when uncertainty sampling can be a poor strategy for classification.

Figure 2.3 illustrates the problem for a binary linear classifier using uncertainty sampling. Shaded polygons represent labelled instances in $\mathcal{L}$, and circles represent unlabelled instances in $\mathcal{U}$. Since A is on the decision boundary, it would be queried as the most uncertain. However, querying B is likely to result in more information about the data distribution as a whole.

Example:

Sample	$\phi_A(x)$	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
x_1	7	6	10	4	8	6	7	3	1	10.780
x_2	7	1	4	4	10	5	9	4	1	10.334
x_3	7	8	3	2	2	1	8	6	3	9.975
x_4	6	4	3	7	8	6	1	5	3	8.798
x_5	4	8	1	5	4	1	7	7	10	6.090
x_6	8	2	5	7	5	10	3	5	7	12.251
x_7	10	3	8	2	4	2	5	5	3	**14.142**
x_8	2	7	2	9	3	3	5	3	1	2.850

$(1) = sim(x, x^{(1)})$, $(2) = sim(x, x^{(2)})$, $(3) = sim(x, x^{(3)})$, $(4) = sim(x, x^{(4)})$,
$(5) = sim(x, x^{(5)})$, $(6) = sim(x, x^{(6)})$, $(7) = sim(x, x^{(7)})$, $(8) = sim(x, x^{(8)})$.
$(9) = \phi_A(x) \times \left(\frac{1}{U}\sum_{u=1}^{U} sim(x, x^{(u)})\right)^{\beta}$
$\beta = 0.25$

In this example it is found that the sample that would be most informative is x_7.

3 Background

This chapter is divided into two sections. Section 1 discusses about Support Vector Machine (SVM) which is the ML methodology employed in this work. Part 2 is devoted to n-gram features.

3.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the most well-known and successfully ML algorithm used in text categorisation and much other fields. In this section, some preliminary definition related to ML concepts and, in particular, to the SVM are given.

Supervised learning is one of the ML tasks aiming to find a discriminant function which relates input patterns or vectors with the corresponding label. In order to do so a training data set made of a set of patterns and their associated categories wherein a discriminant function is obtained. This discriminant function must allow the correct classification of new unknown data afterwards introduced and generated with the same training data distribution which is previously known.

Training data is a set of input data and a corresponding category. Supervised ML algorithms provide a training data set analysis and generate a discriminant function which is used for classifying new samples. In an ideal scenario the ML algorithm allows to detect correctly each new unknown data category afterwards introduced.

SVMs were developed by Vapnik and his collaborators in the 90th[12]. It is one of the best theoretically motivated ML algorithm, based on Statistical Learning theory. Initially SVMs were designed to solve binary classification problems, only later they have been extended to solve multiclass and regression problems as well. The present work focuses on classification tasks, concretely the developed system find out whether a piece of text from a KIID refers to the percentage of the exit charge of the found, so, it performs a binary classification task. Following the usual notation, the two categories are represented by $+1$ and -1. Positive samples correspond to a piece of text that refers to the exit charge percentage and negative samples to a piece of text that refers to other aspects of the fund.

A binary classification problem consists of finding the discriminant function f that allows to assign any pattern or input vector to one of the two categories represented by:

$$f: R^d \rightarrow \{-1, 1\} \tag{3.1}$$

In this work, positive samples such as the exit charge percentage are labelled as 1 and any other word or set of words is labelled as -1. The data set is made of n vectors $x_1, x_2, ..., x_n \in R^d$ with each corresponding label $y_1, y_2, ..., y_n \in \{-1, 1\}$:

$$(x_1, y_1), ..., (x_n, y_n) \tag{3.1.2}$$

SVMs are hyperplane based classifiers which divide the space R^d into two different regions, one for each class. Given a linearly separable dataset, according to the Statistical Learning theory[12], the hyperplane based classifier that minimises the structural risk is the one which maximises the margin between both classes.

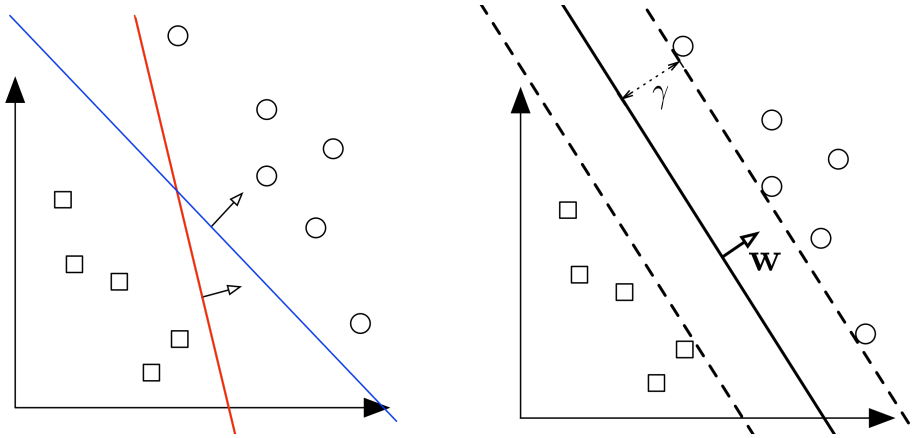


Figure 3.1: Margin maximisation example.

Figure 3.1 illustrates an example of margin maximisation. Left figure shows in red a hyperplane which separates both classes and in blue another hyperplane which divide both classes as well but differently. Right figure shows the hyperplane which maximises the margin between both classes.

Most of methods focus on minimising the model error caused by the training sample, conversely SVM method lie in minimise the structural risk. The general idea is to select an equidistant separable hyperplane between all the closest patterns in the training set to achieve the maximum margin in each side of the hyperplane. In this manner, the optimum hyperplane is defined by the bordering samples of each class. These samples give the method's name, they are called *support vectors*.

The optimal separable hyperplane is defined by a vector $\boldsymbol{w} \in R^d$ and a scalar term b. These terms are found solving an optimisation problem where the hyperplane margin with respect to a training set based on the canonical hyperplanes is maximised. Canonical hyperplanes are those that fulfill the following property:

$$min_{x_i} y_i(\boldsymbol{w^{*T}} \cdot \boldsymbol{x_i} + b^*) = 1 \tag{3.1.3}$$

The normal vector to the canonical hyperplane $\boldsymbol{w^*}$ and the independent term b^* are found through the following normalisation: $\boldsymbol{w^{*T}} = \frac{\boldsymbol{w}}{c}$ and $b^* = \frac{b}{c}$, where $c = min_{x_i} \mid \boldsymbol{w^{*T}} \cdot \boldsymbol{x_i} + b^* \mid$.

The optimisation problem formulation to find the terms $\boldsymbol{w}^*$ and b^* that maximising the margin corresponds to maximize the hyperplane distance to the closest sample, what is equivalent to:

$$\frac{\boldsymbol{w}^{*T} \cdot x + b^*}{\|\boldsymbol{w}^*\|} = \frac{1}{\|\boldsymbol{w}^*\|} \tag{3.1.4}$$

because of the canonical hyperplane property shown in equation 3.1.3. In figure 3.2 the maximised margin between both classes is depicted.

In a linearly separable training sets the last SVM formulation is equivalent to maximising the expression 3.1.4 which is equivalent to:

$$\min_{\boldsymbol{w} \in R^d} \frac{1}{2} \|\boldsymbol{w}\| \tag{3.1.5}$$

$$\text{given that } y_i(\boldsymbol{w_i^T} \cdot x_i + b) \geq 1$$

This problem can be solved applying the Lagrange approach where $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i}$ in such a way that $\boldsymbol{w}$ is a linear combination of input vectors. The dual optimization problem is defined via multipliers as follows:

$$\max_{\alpha_i} L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T x_j} \tag{3.1.6}$$

$$\text{where } \alpha_i \geq 0 \ \forall_i = 1, ..., n$$

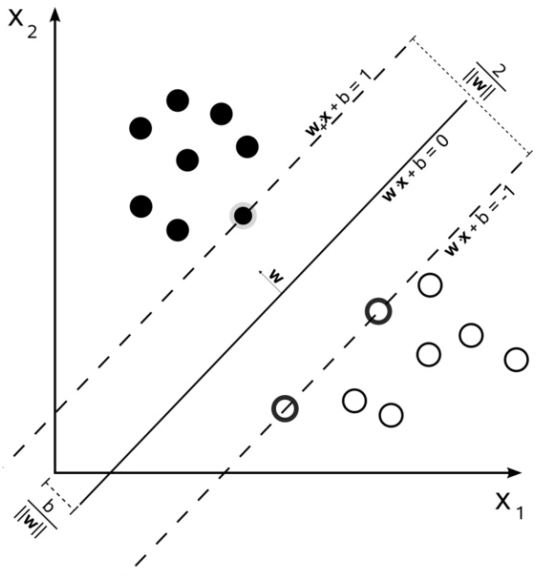


Figure 3.2: Margin maximization optimisation problem.

Input vector which have an $\alpha_i \neq 0$ are called *support vectors*$\{\boldsymbol{s_i}\}$. Those vectors are the only ones used to the linear combination that defines the vector $\boldsymbol{w}$, and belongs to the classification margin. Finally, the discriminant function which will be applied to the new samples $\boldsymbol{x'}$ is:

$$f(\boldsymbol{x'}) = \begin{cases} 1 & if \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i^T} \boldsymbol{x'} + b > 0 \\ -1 & if \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i^T} \boldsymbol{x'} + b < 0 \end{cases} \tag{3.1.7}$$

Up to this point it was supposed that the problem was linearly separable. Although when the problem is not linearly separable is because there is not any separable hyperplane or discriminant function, which allows to distinguish between samples placed in non linear regions. One of the possible solutions to address this problem is to project the original data into a higher dimension space through a function:

$$\Phi : R^d \rightarrow R^m \tag{3.1.8}$$

$$\text{where } m > d$$

Consequently, if the separable hyperplane is $f(\boldsymbol{x}) = \boldsymbol{w^T} \cdot \boldsymbol{x} + b$, in the case of non linearly separable problem it becomes $f(\boldsymbol{x}) = \boldsymbol{w^T} \cdot \Phi(\boldsymbol{x}) + b$, obtaining a linear separable hyperplane in the projected space for all the non linearly separable data.

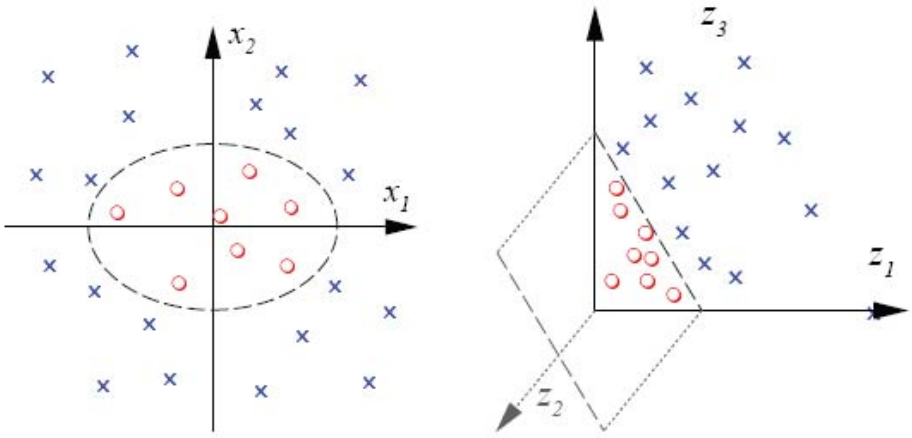


Figure 3.3: High dimension projection space $\Phi : R^2 \rightarrow R^3$.

In order to solve this problem in a computationally efficient manner, it is used the *kernel trick*. With this tool, it is not necessary to know the input data vector representation in the high dimension space. Only the kernel function is required, in such a way that the result of the scalar product in the high dimension space is known from the expression of the low dimension space vectors.

$$k(\boldsymbol{x_1}, \boldsymbol{x_2}) = \varphi(\boldsymbol{x_1})^T \cdot \varphi(\boldsymbol{x_2}) \tag{3.1.9}$$

Most used kernels are:

$$RBF: \quad k(\boldsymbol{x}_1, \boldsymbol{x}_2) = exp(-\gamma \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2) \tag{3.1.10}$$

$$Polynomial: \quad k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \cdot \boldsymbol{x}_2 + h)^p \tag{3.1.11}$$

$$Linear: \quad k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \cdot \boldsymbol{x}_2 \tag{3.1.12}$$

Finally, the classification functions are defined as follows:

$$f(x') = \begin{cases} 1 & if \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x'}, \boldsymbol{x}) + b > 0 \\ -1 & if \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x'}, \boldsymbol{x}) + b < 0 \end{cases} \tag{3.1.13}$$

3.2 SVM Probabilistic Output

Most classifiers output a score of how likely an observation is to be in the positive class. Usually these scores are between 0 and 1 and are called *probabilities*. However these probabilities often do not reflect reality, e.g. a probability of 20% may not mean it has a 20% chance of happening. Platt scaling[13] is a method used to transform classification model outputs into probability distributions over classes. The method was invented by John Platt in the context of SVM[12], replacing an earlier method by Vapnik.

Platt scaling works by fitting a logistic regression model to a classifier's scores. For instance, consider the binary classification problem: we want to determine if input x belong to one of two classes $\{-1, 1\}$. We assume that the classification problem will be solved by a discriminant function f, by predicting a class label $y = sign(f(x))$. For many problems, it is convenient to get a probability $P(y = 1|x)$, i.e. a classification that gives a degree of certainty about the answer. Some classification models do not provide such a probability, or give poor probability estimates.

Platt scaling is an algorithm to solve the aforementioned problem. It produces probability estimates:

$$P(y = 1|x) = \frac{1}{1 + exp(Af(x) + B)} \tag{3.2.1}$$

i.e., a logistic transformation of the classifier scores $f(x)$. A and B are two scalar parameters that are learned by the algorithm. They are estimated using a maximum likelihood method that optimizes on the same training set as that for the original classifier f. Note that predictions can now be made according to $y = 1$ if $P(y = 1|x) > \frac{1}{2}$; if $B \neq 0$, the probability estimates contain a correction compared to the old decision function $y = sign(f(x))$.

Platt additionally suggests transforming the labels y to target probabilities:

$$\text{Positive samples } (y = 1): \qquad t_+ = \frac{N_+ + 1}{N_+ + 2} \tag{3.2.2}$$

$$\text{Negative samples } (y = -1): \qquad t_- = \frac{1}{N_- + 2} \tag{3.2.3}$$

Here, N_+ and N_- are the number of positive and negative samples, respectively.

4 Database description

In this chapter the previous considerations taken into account when working on the database are presented. The considerations are documents for classification and data conditioning.

4.1 Data description

Documents used in this project are KIIDs in **PDF** format obtained from investment companies. The KIID is a two-page 'fact-sheet' style document which includes the critical information about a fund. The document aims to help investors understand the nature and key risks of the fund in order to make a more informed investment decision. For the three classification problems 220 documents were used. Table 4.1 shows the number of positive and negative samples used for each classification problem.

KIID follows a prescribed format. It comprises several sections, providing investors with important information on the fund to help determine whether it aligns with their investment goals, time horizon and risk tolerance:

- **Objectives and Investment Policy**: The investment goal of the fund is among the first key details provided within the KIID. This allows the prospective investor to become acquainted with the fund before they decide to invest, and ensure that the objectives of the fund match their personal investment needs.
- **Risk and Reward Profile**: The profile is an overview of the key risks investors may encounter by investing in a given fund. The Synthetic Risk and Reward Indicator (SRRI) displays the historic volatility of the performance of the fund and categorises it accordingly. The values will range from 1 to 7, where 1 means lower risk and 7 indicates that the level of risk is relatively high. It is important to understand that SRRI is not static as it will be calculated on an on-going basis using the most recent data from the fund.
- **Charges**: KIIDs details only maximum quoted charges. These are shown at a generic level and are not specific to terms negotiated with the adviser.
- **Past Performance**: Past performance is presented in the form of a bar chart showing up to ten years of past performance data. This performance is calculated at the end of each calendar year. If it is relevant relevant, benchmark data will also be shown.

- **Practical Information**: This section includes relevant information such as contact details and where to find further information.

KIIDs can be found in several forms. If the investor uses an intermediary, he or she is the responsible of providing the appropriate KIIDs. Otherwise the investment is done without any intermediary. KIIDs are provided by the invested company or can be found in websites. The aim of the KIID is providing further clarification to the facts and helping the investor to find out more about whether a fund could meet his investment goals.

The database used in this work is made of pieces of text from KIIDs. Each piece of text is labelled according to its category and contains its surrounding information in the document. The obtained database allows to develop a ML algorithm capable to solve specific problems in KIIDs such as identifying *exit charge, fund launch date or available document languages*. An *exit charge* is a fee charged to investors when they redeem shares from a fund. *Fund launch date* informs about the date on which the fund began its operations. And *available document languages* specify in which languages KIIDs can be requested.

In order to solve these problems, identifying *exit charge*, *fund launch date* or *available document languages*; several KIID will be labelled based on different features such as:

- Content: Content of the sample.
- Left context: Sentence at the left side of the sample.
- Right context: Sentence at the right side of the sample.
- Bottom context: Sentence at the bottom side of the sample.
- Section title: Title of the section the sample belongs to.

For instance, in order to identify *exit charge* percentage that are pieces of text which contains exit charge values. Pieces of text that contains *exit charge* percentage have been labelled as positive samples and pieces of text of the same document which does not contain any exit charge information and have no relation with exit charge sentences have been labelled as negative samples. Both are added to the database with its surrounding information.

4.2 Data extraction

There are several reasons why extracting data from PDF can be challenging, starting from technical issues till practical workflow drawbacks. Documents are easily readable for humans, while computers are not capable to understand. For instance, the computer needs to apply an Optical Character Recognition (OCR) technique for scanned image text. Once it is applied, it is possible to manually work with parts of the text. Obviously, this method is tedious. It needs to open each PDF, locate and select the interesting text

and copy it to another software. It takes too much time.

In this work, data extraction is performed through a specific labelling application developed by the company where I am currently doing this project. This application is located in a server and is used through a web browser. The procedure followed to build the database made of pieces of text from KIIDs as follows:

1. Uploading the KIIDs in PDF format to the application.
2. Selecting the classification problem, i.e. *exit charge* detection.
3. Selecting a KIID.
4. Labelling the positive samples of this KIID, i.e. squaring the positive samples. In figure 4.1 it is shown as a *fund launch date* labelling.
5. Splitting the rest of the KIID into pieces of text which represent the negative samples of the database.
6. Proceeding with the next KIID.

Once these steps are completed, the database can be downloaded from the application. The database is structured in the following manner. It contains as many rows as positive and negative samples have been labelled. Each column of the database corresponds to each feature of the sample, label and *document Id*. *Document Id* corresponds to the name of the document which each sample belongs.

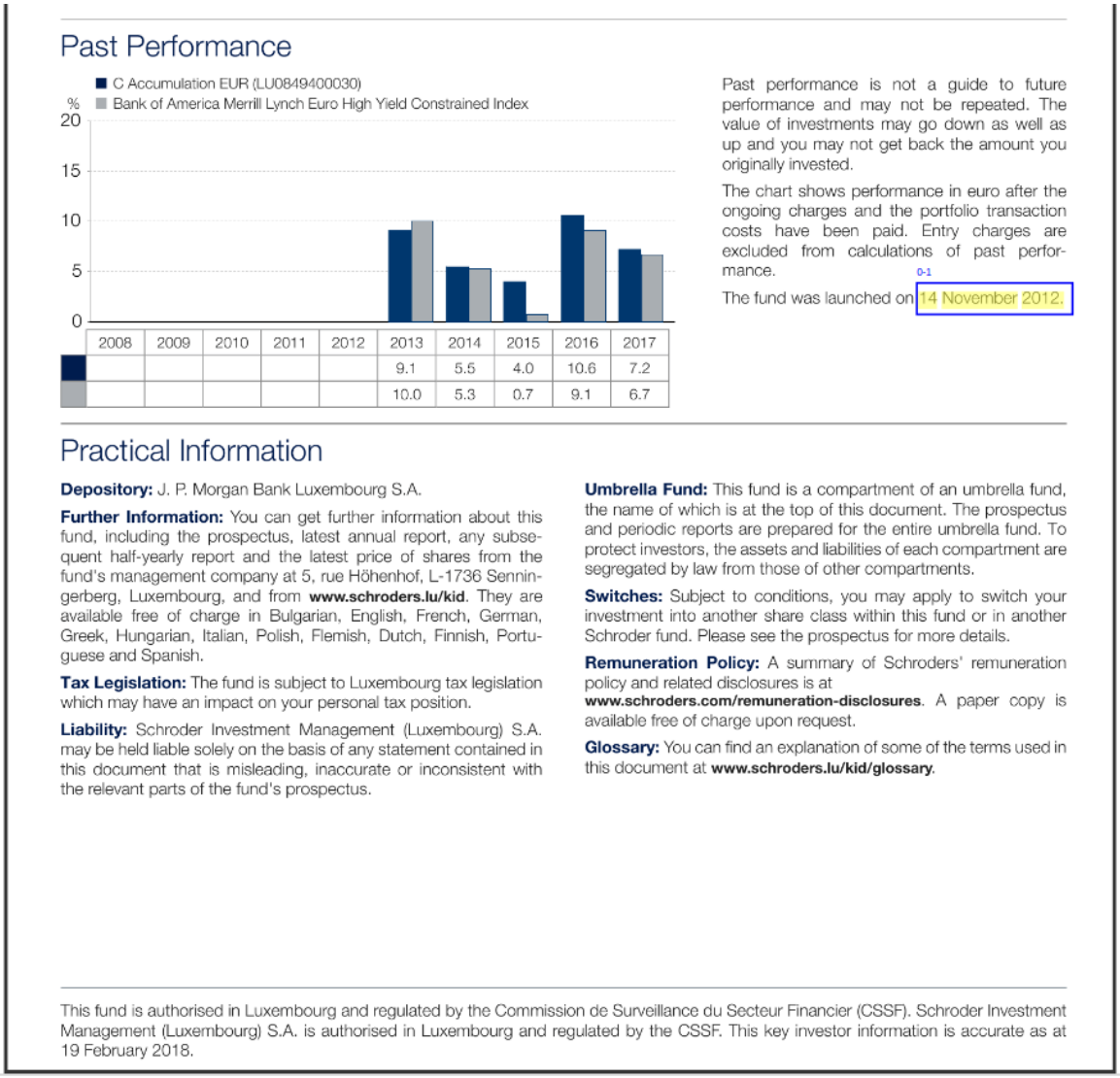
Figure 4.1 illustrates an example of *fund launch date* annotation. The fund launch date is shown in a blue rectangle. Figure 4.2 shows in blue an *exit charge* annotation in the KIID document and figure 4.13 illustrates an example of *available document languages* annotation. In this figure several annotations in colour rectangles are found. These are positive samples for exit charge fund launch date and available document languages identification problems.

	Fund launch date	*Exit charge*	*Available document language*
KIIDs	220	220	220
+ Samples	203	212	248
− Samples	29186	18477	28812

Table 4.1: Database length.

In this project three different problems were studied: identifying *exit charge*, *fund launch date* and *available document languages*. In table 4.1 the length of each database is noted and the number of KIIDs, positive samples and negative samples in the database. In some classification problems the number of positive samples is less than the number of KIIDs, this is because sometimes the required information does not appear in the document. For instance, the fund launch date information is missing in the document because the fund is not launched yet.

In tables 4.2 and 4.3 an example of positive and negative *fund launch date* samples extracted from KIIDs using the labelling tool is depicted.



Past Performance

C Accumulation EUR (LU0849400030)
Bank of America Merrill Lynch Euro High Yield Constrained Index

%
20
15
10
5
0

	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
						9.1	5.5	4.0	10.6	7.2
						10.0	5.3	0.7	9.1	6.7

Past performance is not a guide to future performance and may not be repeated. The value of investments may go down as well as up and you may not get back the amount you originally invested.

The chart shows performance in euro after the ongoing charges and the portfolio transaction costs have been paid. Entry charges are excluded from calculations of past performance.

The fund was launched on 14 November 2012.

Practical Information

Depository: J. P. Morgan Bank Luxembourg S.A.

Further Information: You can get further information about this fund, including the prospectus, latest annual report, any subsequent half-yearly report and the latest price of shares from the fund's management company at 5, rue Höhenhof, L-1736 Senningerberg, Luxembourg, and from **www.schroders.lu/kid**. They are available free of charge in Bulgarian, English, French, German, Greek, Hungarian, Italian, Polish, Flemish, Dutch, Finnish, Portuguese and Spanish.

Tax Legislation: The fund is subject to Luxembourg tax legislation which may have an impact on your personal tax position.

Liability: Schroder Investment Management (Luxembourg) S.A. may be held liable solely on the basis of any statement contained in this document that is misleading, inaccurate or inconsistent with the relevant parts of the fund's prospectus.

Umbrella Fund: This fund is a compartment of an umbrella fund, the name of which is at the top of this document. The prospectus and periodic reports are prepared for the entire umbrella fund. To protect investors, the assets and liabilities of each compartment are segregated by law from those of other compartments.

Switches: Subject to conditions, you may apply to switch your investment into another share class within this fund or in another Schroder fund. Please see the prospectus for more details.

Remuneration Policy: A summary of Schroders' remuneration policy and related disclosures is at **www.schroders.com/remuneration-disclosures**. A paper copy is available free of charge upon request.

Glossary: You can find an explanation of some of the terms used in this document at **www.schroders.lu/kid/glossary**.

This fund is authorised in Luxembourg and regulated by the Commission de Surveillance du Secteur Financier (CSSF). Schroder Investment Management (Luxembourg) S.A. is authorised in Luxembourg and regulated by the CSSF. This key investor information is accurate as at 19 February 2018.

Figure 4.1: Illustration of fund launch date annotation.

Input Feature	**Content**	**14 November 2012**
	Paragraph	The fund was launched on 14 November 2012
	Left context	The fund was launched on
	Right context	∅
	Bottom context	∅
	Section title	Past performance
Output feature	Class label	1
	Document Id	LU0295110042

Table 4.2: Fund launch date positive pattern.

Input Feature	**Content**	**19 February 2016**
	Paragraph	This fund is authorised (...) February 2016.
	Left context	∅
	Right context	∅
	Bottom context	∅
	Section title	∅
Output feature	Class label	0
	Document Id	LU0295110042

Table 4.3: Fund launch date negative pattern.

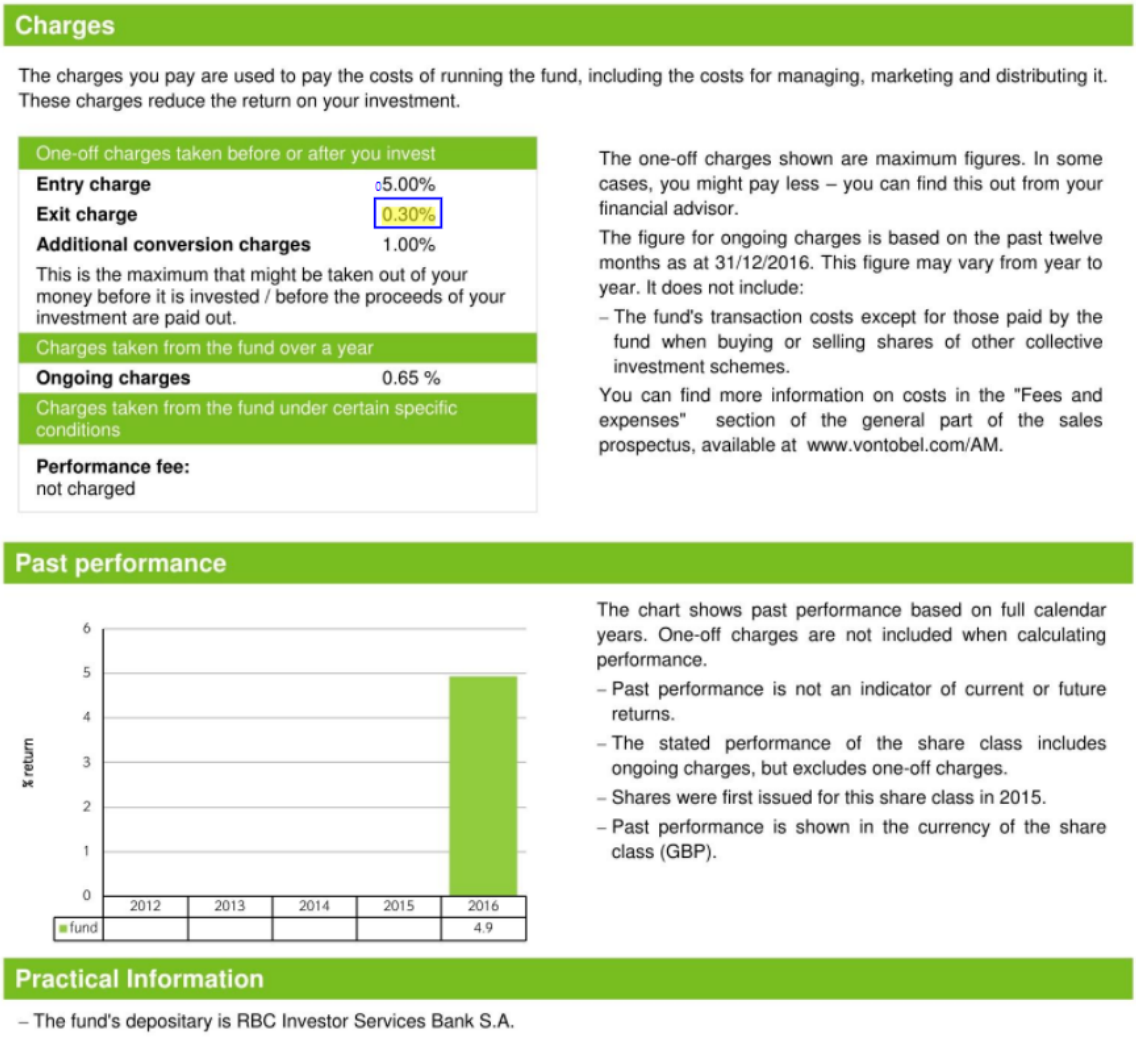


Figure 4.2: Illustration of exit charge annotation.

Input Feature	**Content**	**0.30**%
	Paragraph	**0.30**%
	Left context	Exit charge
	Right context	The one-off (...) financial advisor.
	Bottom context	Additional conversion charges 1.00%
	Section title	Charges
Output feature	Class label	1
	Document Id	FR0056487458

Table 4.4: Exit charge positive pattern

Input Feature	**Content**	**0.65**%
	Paragraph	**0.65**%
	Left context	Ongoing charges
	Right context	The fund's (...) investment schemes.
	Bottom context	Charges taken (...) specific conditions
	Section title	Charges
Output feature	Class label	0
	Document Id	FR0056487458

Table 4.5: Exit charge negative pattern

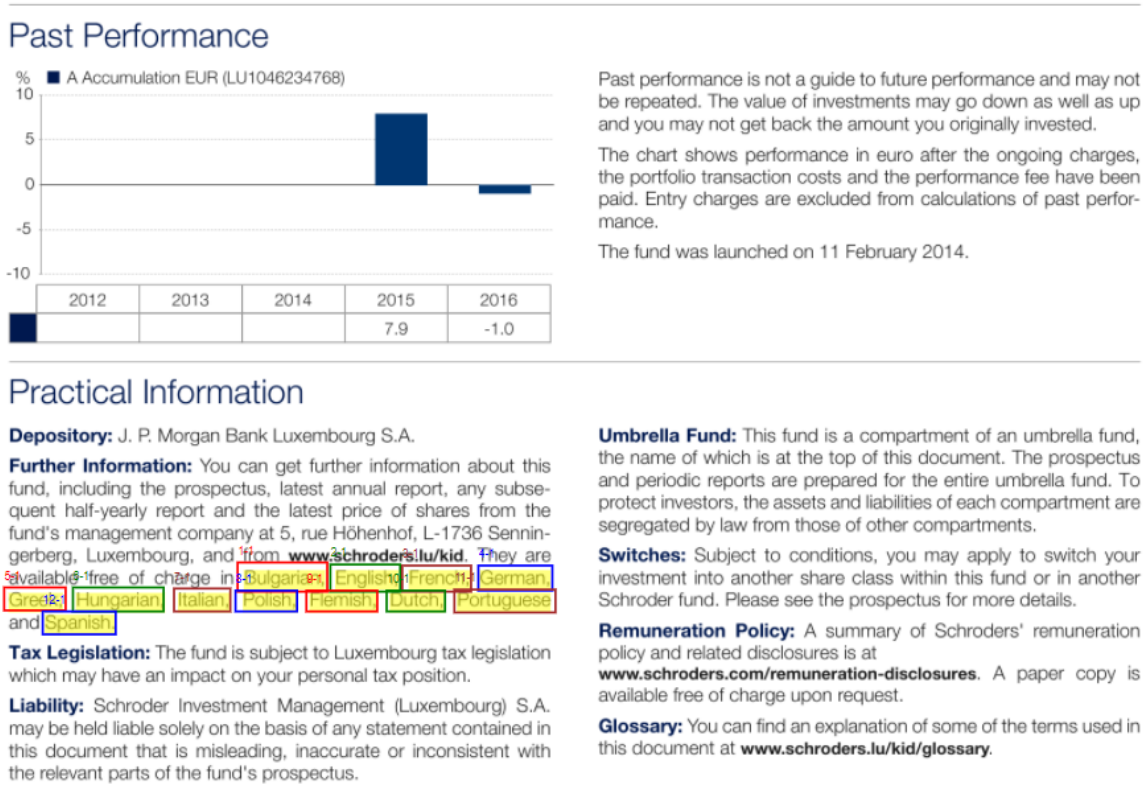


Figure 4.3: Illustration of available languages annotation.

Sometimes KIIDs contain several positive annotations. In figure 4.13 12 different annotations corresponding to: Bulgarian, English, French, and others are found. The following tables show the classification database content for a positive sample and a negative sample in *available document languages*.

	Content	**French**
Input Feature	Paragraph	Further information: (...), French, (...) and Spanish.
	Left context	∅
	Right context	Switches: (...) for more details.
	Bottom context	Tax legislation: (...) tax position.
	Section title	Practical Information
Output feature	Class label	1
	Document Id	LU2254476125

Table 4.6: Available languages positive pattern (French).

	Content	**Schroder**
Input Feature	Paragraph	Liability: Schroder (...) fund's prospectus.
	Left context	∅
	Right context	Remuneration Policy: (...) upon request.
	Bottom context	∅
	Section title	Practical Information
Output feature	Class label	0
	Document Id	LU2254476125

Table 4.7: Available languages negative pattern.

4.3 Data conditioning

SVM is a supervised ML algorithm which can be used for both classification or regression challenges. However, in this work it is used in a classification problem. SVM needs numerical data in order to be trained. It cannot work with text data so it is needed to find a way to convert textual information into numerical or transform the data in order that the SVM becomes capable to work with it. There are several ways to transform text information into numerical. Bag-of-words model approach[14], N-gram[15] or even binary features[16] where 1 or 0 represent the presence or absence of a word in the text.

In this work, the data transformation is done through the aforementioned methods where data used are pieces of text and its surrounding information extracted from KIIDs. In order to train the ML algorithm data needs to be conditioned and finally converted into a numerical feature vectors. In this section it is presented how the data was conditioned in order to train the ML algorithms.

1. Generating the vocabulary: creating a dictionary of terms present in positive samples. To do so:
   - Converting the text to lower case: *Fund* or *fund*, it does not matter, so all the text can be brought to lower case.
   - Removing *Stop words*: words like *a* or *the* should not be so significant. These words are known as *Stop Words*, and are removed while conditioning the data.
   - Lemmatizing words: *invest* and *invested* would suggest a similar meaning, so lemmatization is done to bring the word in root form. *The Stanford CoreNLP Natural Language Processing Toolkit*[17] is used in this project to do the lemmatization.

Vocabulary	Words				
	Paragraph	Left context	Right context	Bottom context	Section title
1	additional	fund	charges	bond	objective
2	additional information	fund management	charges from	bond fund	charges
3	funds	higher	fund	counterparty	performance
4	funds risk	higher possible	fund launch	counterparty risk	past performance
5	..	...	...	...	...
...	volatility	worldwide	shares	purchase	risks

Table 4.8: *Vocabulary example.*

Table 4.8 shows an example of a condensed vocabulary used in this project. As shown in this table, vocabulary contains either one word (*Uni*-gram) or groups of two words (*Bi*-grams). Words are added to the vocabulary if they appear at least 20 times in the positive samples of the database.

Each classification problem, identifying *exit charge*, *fund launch date* or *available document languages* has his own vocabulary. In table 4.9 the length of each vocabulary's problem is shown.

2. Creating the feature vectors: Each sample of the database is converted into a vector space. Each term of the vector represents the presence or the absence of the

Problem	Vocabulary length					
	Paragraph	Left context	Right context	Bottom context	Section title	Total
Fund launch date	1760	994	1860	2316	8	6938
Exit charge	1743	455	1260	1062	8	4528
Available document language	1758	992	1860	2316	8	6934

Table 4.9: *Vocabulary lengths.*

vocabulary words in the sample which is going to be classified. 1 if the vocabulary appears in the sample or 0 if not. A *term-frequency* can be used to represent each term as well. *Term-frequency* is a measure of how many times the terms present in the indexed vocabulary appear in the data set samples. In this case, *fund launch date, exit charge* and *available document language* detection are very simple problems by considering the presence or absence of vocabulary terms. Each feature vector length is defined according to its vocabulary length. Last column of table 4.9 shows the length of the feature vector for each classification problem. For instance, the feature vector length of the *fund launch date* is 6938.

For example, let's take the piece of text below to define our document space:

- Positive samples:
  - p1: *The fund launch date*
  - p2: *The fund was launched in*
- Test set:
  - d3: *The fund was launched in December 1997.*
  - d4: *The share class launch date was*

First the vocabulary is created from the positive samples. Text is converted to lower case, stop words are removed and the vocabulary is brought to its root. Vocabulary becomes *fund, launch* and *date*. Second the test document set is converted into vector space where each term of the vector represents the presence or absence of the vocabulary in the sample. The first term of the vector represents the word *fund* of the vocabulary, the second represents *launch* and so on.

In this work 5 different features are used: *Content, left context, right context, bottom context and section title*. Each feature has its own vocabulary extracted from positive samples which contains *Uni*-grams and *Bi*-grams. For instance, the feature vectors used to train the ML algorithm for each problem have been structured as follows:

	fund	*launch*	*date*	Output Class
d3	1	1	0	1
d4	0	1	1	0

Table 4.10: *Feature vectors example.*

- *Fund launch date* detection:

Piece of text: *14 November 2012*

	***Uni*-gram**														
Feature	**Content**			**Left context**			**Right context**			**Bottom context**			**Section title**		
Vocabulary	*November*	*class*	...	*Launched*	*investment*	...	*chart*	*shares*	...	*Description*	*Fund*	...	*objective*	*performance*	...
Fueature vector	1	0	...	1	0	...	0	0	...	0	0	...	0	1	...

	***Bi*-gram**														
...	**Content**			**Left context**			**Right context**			**Bottom context**			**Section title**		
...	*14 November*	*Share class*	...	*was launched*	*your investment*	...	*chart shows*	*your shares*	...	*<s> Description*	*Fund is*	...	*Objective about*	*Past performance*	...
...	1	0	...	1	0	...	0	0	...	0	0	...	0	1	...

Table 4.11: *Fund launch date* feature vector example.

- *Exit charge* detection:

Piece of text: *0.30%*

	Uni-gram														
Feature	**Content**			**Left context**			**Right context**			**Bottom context**			**Section title**		
Vocabulary	*0,30*	*class*	...	*Exit*	*investment*	...	*financial*	*fund*	...	*Additional*	*tax*	...	*Charges*	*performance*	...
Fueature vector	1	0	...	1	0	...	1	0	...	1	0	...	1	0	...

	Uni-gram														
...	**Content**			**Left context**			**Right context**			**Bottom context**			**Section title**		
...	*0,30%*	*share class*	...	*Exit charge*	*your investment*	...	*financial advisor*	*fund of*	...	*Additional conversion*	*your tax*	...	*Charges </s>*	*Past performance*	...
...	1	0	...	1	0	...	1	0	...	1	0	...	1	0	...

Table 4.12: Exit charge feature vector example.

- *Available document language* detection:

Piece of text: *French*

	Uni-gram													
Feature	**Content**			**Left context**			**Right context**			**Bottom context**			**Section title**		
Vocabulary	*French*	*Share*	...	*Exit*	*investment*	...	*financial*	*fund*	...	*Additional*	*Tax*	...	*Practical*	*performance*	...
Fueature vector	1	0	...	0	0	...	0	0	...	0	1	...	1	0	...

	Uni-gram													
...	**Content**			**Left context**			**Right context**			**Bottom context**			**Section title**		
...	*French, German*	*Share class*	...	*Exit charge*	*yout investment*	...	*financial advisor*	*fund launch*	...	*Additional conversion*	*Tax legislation*	...	*Practical information*	*Past performance*	...
...	1	0	...	0	0	...	0	0	...	0	1	...	1	0	...

Table 4.13: Available document language feature vector example.

5 Experiments

This chapter describes the implementation of the active learning strategies studied. Three different active learning query strategies were chosen:

- Uncertainty sampling
- Query by Committee
- Density-Weighted method

These active learning strategies are compared with the passive learning methodology.

5.1 Environment

The following software, respectively operating systems and libraries, were used for the implementation:

- Ubuntu 16.04 LTS
- Java "1.8.0.161"
- IntellyJ IDEA 2017.3.4
- Statistical Machine Intelligence and Learning Engine (SMILE)
- The Stanford CoreNLP Natural Language Processing Toolkit
- Annotation tool

5.2 Learning process

As explained in section 3.1, SVM needs a training phase. For training a 10-Fold Cross-Validation (CV) was used. 10-Fold CV is a cross-validation technique used to evaluate the results and guarantee that the validation and training data are independent. In order to apply the 10-Fold CV, the database have been split in 10 groups, shown in table 4.1. In this work 220 KIIDs were used . In order to accomplish the training phase. 22 KIIDs were used for training another 22 KIID(s) have been used for validation and the rest of the 176 documents were used as an unlabelled pool of samples.

Figure 5.1 shows in grey the samples used for training, in yellow those used as validation and in green the rest of data used as the unlabelled pool of data. Each cell of the matrix represents a group of 22 KIID(s). For each new training set generated through

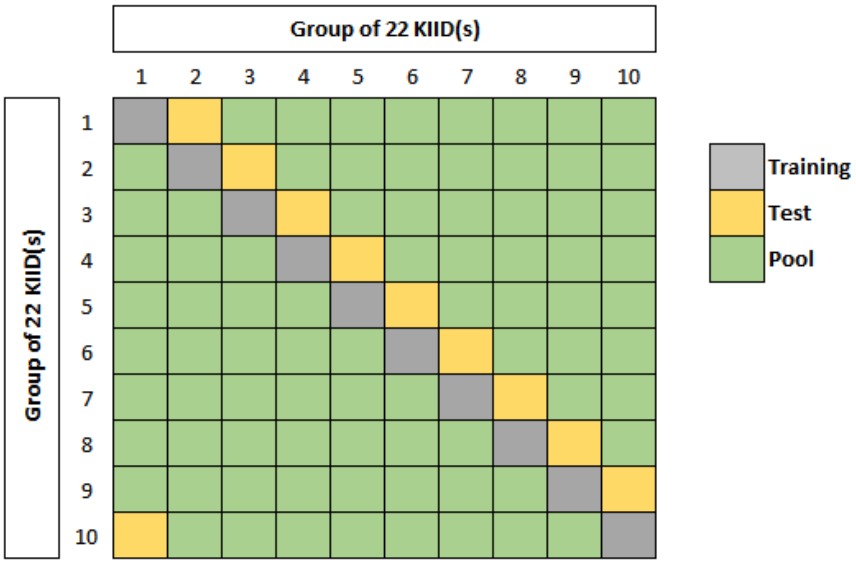


Figure 5.1: 10-Fold CV process.

10-Fold CV a model with which the validation set is evaluated is obtained.

The following sections describes both methodologies: AL and PL. It can be seen that the initial model calculation is the same for both methodologies, the difference lies in the selection of new training samples for the model update.

5.3 Passive learning

In this section, the passive learning methodology used for annotation and classification is described. Theoretically, the PL workflow follows the next steps:

- Calculating the initial model:
  1. Human annotates n samples using the labelling application.
  2. Splitting the data into train, validation and unlabelled pool of data.
  3. Obtaining the initial model.
  4. Validating the initial model.
- Updating the model:
  1. Classifying the unlabelled pool of data.
  2. Selecting randomly k samples. Adding them to the training set and removing them from the unlabelled pool of data.
  3. Training the model using the new updated training set.

4. Validating the updated model.

This methodology does not follow any selection criterion for selecting the new training samples. Selecting the new training samples randomly mainly causes the performance of the system increases slow. Figure 5.2 shows a scheme of passive learning methodology. In this figure there is not any automatised task that helps the user to reduce and facilitate the labelling task.

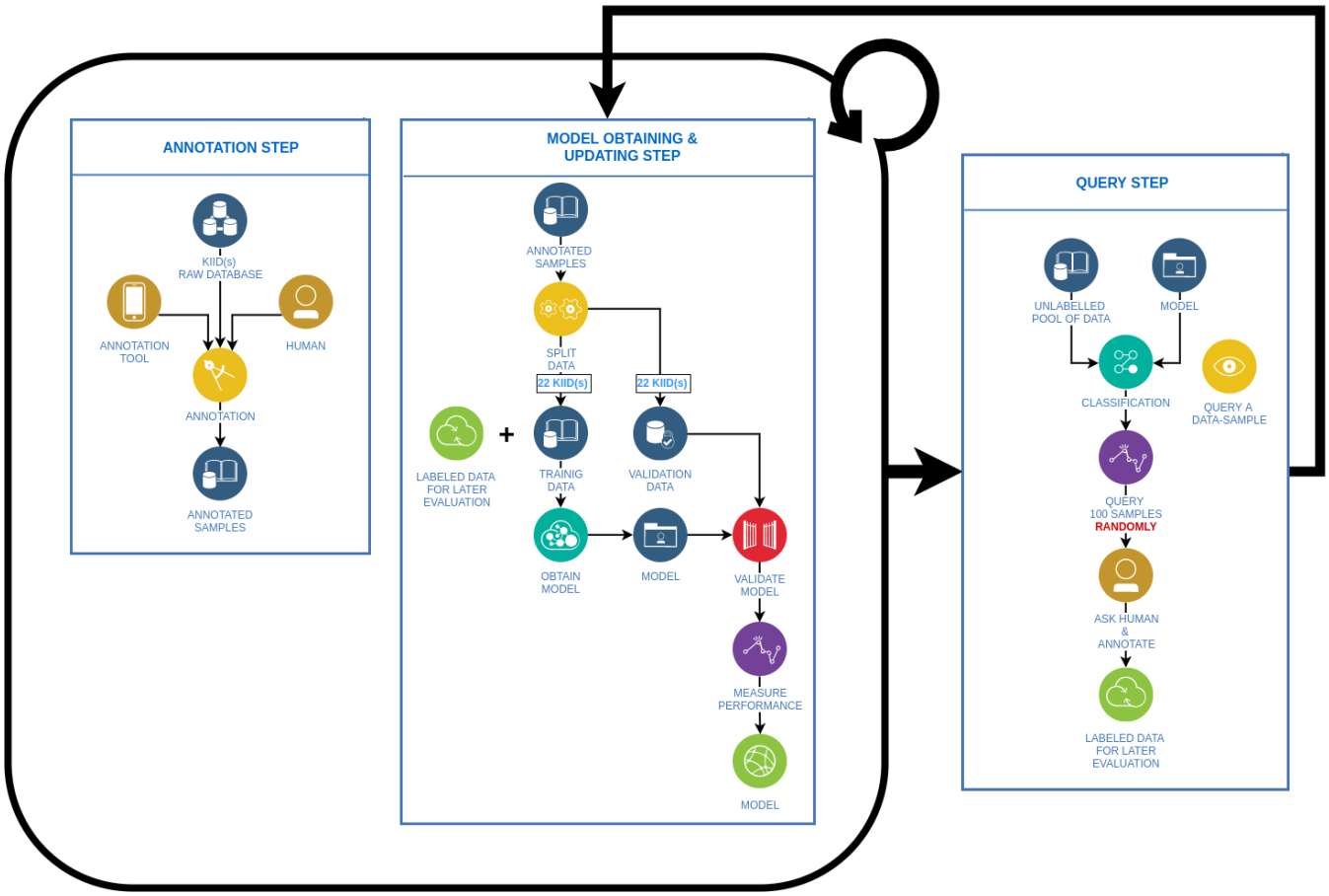


Figure 5.2: Passive learning methodology.

In order to implement the PL methodology the following order of action were undertaken. From the database 22 KIIDs were taken to train an initial model. Another 22 documents, which the real class is known, were used to validate the model and calculate its performance. From the unlabelled pool of data, 100 samples are randomly selected. Using the *document Id* from those 100 samples, all the samples which belongs to that *document Id* are added to the training set and removed from the unlabelled pool of data.

With the new training set, the new model is calculated and validated with the validation set, the same used when validating the initial model.

For instance, imagine that we select 100 samples which belongs to 6 different KIID(s). We select all the samples from the unlabelled pool of data that belongs to that 6 KIID(s) and add them to the training set. Next, we remove those samples from the unlabelled pool of data and train again the model. This way of proceed pretends to imitate the real application workflow where an oracle will annotate those 6 KIID(s) in order to improve the initial model.

5.4 Active learning

In this section the AL approach, the method of selecting new training data, is described. This study focuses on a pool-based sampling scenario. Initially, there is a small set of labelled data and a large static or non-changing pool of unlabelled data. Figure 5.3 shows the active learning proposal used in this project. As stated earlier, the objective of this work is to find a methodology which diminishes the number of samples to be annotated maintaining the performance of the system. In order to do so, three different active learning query strategies are studied:

- Uncertainty sampling.
- Query by Committee.
- Density-weighted method.

Annotation step, shown in figure 5.3, works as a passive learning work-flow. The objective was to obtain an initial model that is improved later with different active learning approaches, labelling only the required samples queried by the system.

The next stage, Model Obtaining & Updating Step, is where the initial model is calculated and through the active learning methodology the model is performed repetitively until a Confidence Interval (CI) is achieved. AL performs over the model adding the queried samples to the training set. AL flow is developed in the following manner:

- Calculate the initial model:
    1. A person annotates n samples using the labelling application.
    2. Splitting the data into train, validation and unlabelled pool of data.
    3. Obtaining the initial model.
    4. Validating the initial model.
- Update the model:
    1. Classifying the unlabelled pool of data.

2. Querying k samples to be labelled according to the previously presented AL selection criterion. Adding them to the training set and removing them from the unlabelled pool of data.
3. Validating the model using the new updated training set.
4. Validating the updated model.

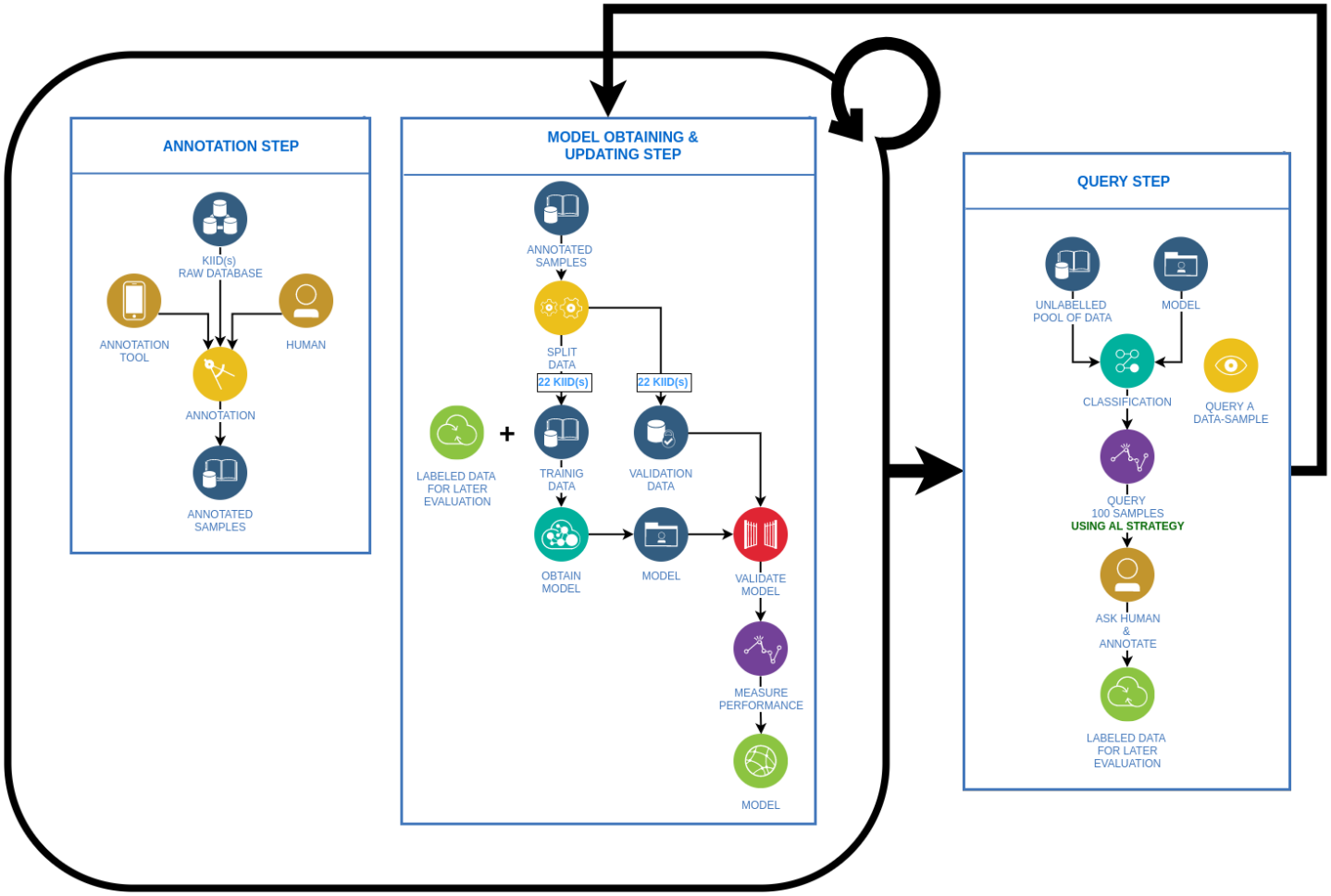


Figure 5.3: Active learning methodology.

The AL methodology was implemented following the PL steps presented in last section. First the 22 KIIDs were selected and train an initial model. The initial model was then validated using another 22 documents and its performance was calculated. From the unlabelled pool of data, using each of the AL selection criterion presented, $k = 100$ samples were selected. Using the *document Id* from those 100 samples, all the samples which belongs to that *document Id* were added to the training set and all the samples

of the documents were removed from the unlabelled pool of data.

5.5 Query strategies

The main difference between active and passive learning is the ability to query instances based upon past queries and their labels. As stated before, all active learning scenarios require to know the informativeness of the unlabelled instances. In this section, the three approaches for querying instances implemented are explained.

5.5.1 Uncertainty Sampling

It is the simplest and most commonly used query. In this framework, AL queries the instances that are least certain on its classification. This approach is straightforward for probabilistic learning models. Furthermore, it can be extrapolated in order to be used through SVM.

Algorithm 1: Uncertainty sampling function

Data: training set, pool set, model, number of samples
Result: (new training set, new pool set, updated model)
while *pool set is NOT empty* **do**
    Classifying the pool set ($\phi_A(x)$);
    Sorting in ascending order according to probability classification;
    Selecting *document ID* of k most uncertain samples;
    Adding all the samples belonging to these *document Id* to the training set;
    Removing all the samples belonging to these *document Id* from the pool;
    Updating the model;
end

Algorithm 1 shows the procedure of the uncertainty sampling function. AL workflow updates the model selecting the most informative samples of the unlabelled pool of data. Uncertainty sampling selects the most informative samples using the probability classification obtained trough SVM. The probability classification results a value between 0.5 and 1. A classification output closer to 1 represents a very accurate classification. On the other hand, a classification output closer to 0.5 means a very inaccurate classification.

Once all the pool is classified, the samples are sorted in ascending order based on the probability classification. The first 100 samples are selected. Using the *document Id.* of these 100 samples all the samples which correspond these *document Id* are added to the training set and removed from the unlabelled pool of data. These procedure is repeated until there are no more samples in the pool of unlabelled data.

5.5.2 Query by Committee

This strategy works by using a committee of models querying those unlabelled examples about which the committee disagrees the most about the label.

Algorithm 2: Query by committee function

Data: training set, pool set, model 1, model 2, number of samples
Result: (new training set, new pool set, updated model)

```
while not at end of the pool set do
    Classifying the pool set with m models;
    Normalising the classification (φ_A(x)) between 0 and 1;
    Calculating the difference between the two model classifications;
    Sorting the difference in descending order;
    Adding all the samples belonging to these document Id to the training set;
    Removing all the samples belonging to these document Id from the pool;
    Updating the model;
end
```

Algorithm 2 shows the procedure of the query by committee sampling function. In this work it has been used $m = 2$ different SVM models with different parameter configuration. This query strategy prioritise the samples in which the two different SVM show more disagreement.

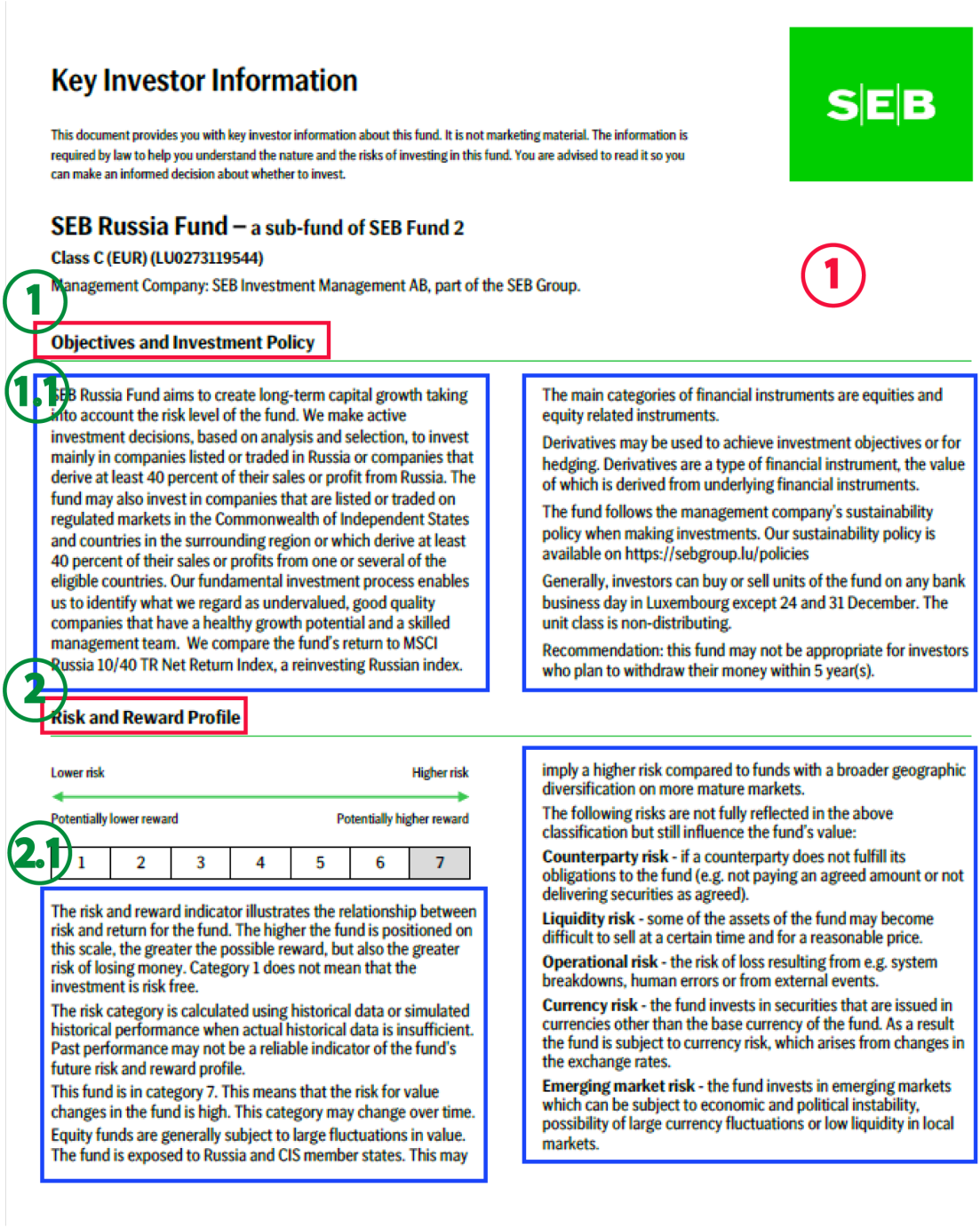
In Query by Committee application, the unlabelled pool of data is classified using the initial model by the two SVMs. In order to select those samples which more disagree in the classification, the output of the SVM of each samples are normalised between 0 and 1.

$$z_i = \frac{x_i - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \tag{5.5.1}$$

The difference of the normalised classification of each SVM output give us the disagreement between classifiers. The higher the result is the bigger the disagreement. This result is sorted in descending order and the first 100 samples are selected. From these 100 samples the *document Id.* is chosen and all the samples corresponding to these *document Id.* are added to the training set and removed from the unlabelled pool of data. This procedure is repeated until the unlabelled pool of data becomes empty.

5.5.3 Density-Weighted method

The main idea of the Density-Weighted method is that informative instances should not only be those which are most uncertain, but also those which are "representative" of the underlying distribution. So, this query strategy takes into account how similar are the



Key Investor Information

This document provides you with key investor information about this fund. It is not marketing material. The information is required by law to help you understand the nature and the risks of investing in this fund. You are advised to read it so you can make an informed decision about whether to invest.

SEB

SEB Russia Fund – a sub-fund of SEB Fund 2

Class C (EUR) (LU0273119544)

Management Company: SEB Investment Management AB, part of the SEB Group.

1

1

Objectives and Investment Policy

1.1

SEB Russia Fund aims to create long-term capital growth taking into account the risk level of the fund. We make active investment decisions, based on analysis and selection, to invest mainly in companies listed or traded in Russia or companies that derive at least 40 percent of their sales or profit from Russia. The fund may also invest in companies that are listed or traded on regulated markets in the Commonwealth of Independent States and countries in the surrounding region or which derive at least 40 percent of their sales or profits from one or several of the eligible countries. Our fundamental investment process enables us to identify what we regard as undervalued, good quality companies that have a healthy growth potential and a skilled management team. We compare the fund's return to MSCI Russia 10/40 TR Net Return Index, a reinvesting Russian index.

The main categories of financial instruments are equities and equity related instruments.

Derivatives may be used to achieve investment objectives or for hedging. Derivatives are a type of financial instrument, the value of which is derived from underlying financial instruments.

The fund follows the management company's sustainability policy when making investments. Our sustainability policy is available on https://sebgroup.lu/policies

Generally, investors can buy or sell units of the fund on any bank business day in Luxembourg except 24 and 31 December. The unit class is non-distributing.

Recommendation: this fund may not be appropriate for investors who plan to withdraw their money within 5 year(s).

2

Risk and Reward Profile

Lower risk — Higher risk

Potentially lower reward — Potentially higher reward

2.1

1	2	3	4	5	6	7

The risk and reward indicator illustrates the relationship between risk and return for the fund. The higher the fund is positioned on this scale, the greater the possible reward, but also the greater risk of losing money. Category 1 does not mean that the investment is risk free.

The risk category is calculated using historical data or simulated historical performance when actual historical data is insufficient. Past performance may not be a reliable indicator of the fund's future risk and reward profile.

This fund is in category 7. This means that the risk for value changes in the fund is high. This category may change over time.

Equity funds are generally subject to large fluctuations in value. The fund is exposed to Russia and CIS member states. This may imply a higher risk compared to funds with a broader geographic diversification on more mature markets.

The following risks are not fully reflected in the above classification but still influence the fund's value:

Counterparty risk - if a counterparty does not fulfill its obligations to the fund (e.g. not paying an agreed amount or not delivering securities as agreed).

Liquidity risk - some of the assets of the fund may become difficult to sell at a certain time and for a reasonable price.

Operational risk - the risk of loss resulting from e.g. system breakdowns, human errors or from external events.

Currency risk - the fund invests in securities that are issued in currencies other than the base currency of the fund. As a result the fund is subject to currency risk, which arises from changes in the exchange rates.

Emerging market risk - the fund invests in emerging markets which can be subject to economic and political instability, possibility of large currency fluctuations or low liquidity in local markets.

Figure 5.4: Document annotation for document feature vector creation.

samples.

In order to take into account the similarity of the samples, a feature vector which represents the similarity between documents has been created. This feature vector is created using the labelling tool used for labelling positive and negative samples. Labelling tool is capable to recognise each section of the document identifying its section title and content. Figure 5.4 shows the annotation task when labelling the documents for document feature vector creation.

The procedure followed in order to obtain the document feature vector is the next:

1. Uploading the KIID(s) in PDF format to the application.
2. Selecting the document classification task.
3. Selecting a KIID.
4. Labelling each section title of the document and next its content. Then proceed with the next section title and section content in the same manner, i.e. squaring first (1) the section title and next (1.1) the section content. In figure 5.4 it is shown an annotated KIID where the red rectangles correspond to the section title and the blue ones to the section content.
5. Proceeding with the next KIID.

As demonstrated before converting the text samples into numerical feature vectors, in order to calculate a similarity measure, the documents are converted into feature vectors. To proceed in the following manner.

1. Generating two different vocabularies:
   - Creating a dictionary of terms present in all documents. A list of all the terms present in all documents.
   - Creating a vocabulary of terms present in each document. A list of terms present only in each document.

   These two different dictionaries of vocabulary will allow us to identify which words or groups of two words present in the feature vector appear in each document. If the word or group of two words appear in the document, those terms of the vector will appear as 1, otherwise it will appear as 0. The vocabulary of all document contains 10435 words and groups of two words. The feature vector length used to calculate the similarity between documents is 10435.
2. Calculating the Euclidean distance between each document feature vector.

$$dist(\mathbf{a}, \mathbf{b}) = dist(\mathbf{b}, \mathbf{a}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2} \qquad (5.5.2)$$

3. Computing the distance matrix. Distance matrix is a symmetric matrix where each element corresponds to the Euclidean distance between two documents. For instance, cell $(1, 2)$ contains the euclidean distance between documents 1 and 2.

$$\begin{bmatrix} dist_{(1,1)} & dist_{(1,2)} & dist_{(1,3)} & dist_{(1,j)} \\ dist_{(2,1)} & dist_{(2,2)} & dist_{(2,3)} & dist_{(2,j)} \\ dist_{(3,1)} & dist_{(3,2)} & dist_{(3,3)} & dist_{(3,j)} \\ dist_{(j,1)} & dist_{(j,2)} & dist_{(j,3)} & dist_{(j,j)} \end{bmatrix} \tag{5.5.3}$$

Algorithm 3: Density-weighted method function

Data: training set, pool set, model, number of samples, distance matrix
Result: (new training set, new pool set)
while *not at end of the pool set* **do**
  Classifying the pool set ($\phi_A(x)$);
  Calculating the Informativeness of each sample
  $Inf = \phi_A(x) \cdot \left(\frac{1}{U}\sum_{u=1}^{U} sim(x, x^{(u)})\right)^{\beta}$;
  Sorting in ascending order according to Informativeness;
  Selecting *document ID* of n most informative samples;
  Adding all the samples belonging to these *document Id* to the training set;
  Removing all the samples belonging to these *document Id* from the pool;
  Updating the model;
end

Distance matrix is a symmetric matrix where the diagonal cells contain 0 because the difference between the same feature vector is 0. In order to operate with the SVM output and the similarity parameter, they need to be normalised between 0 and 1 accordingly. So, SVM output is normalised between 0 and 1 shown in equation 5.5.1. Distance is a measure of dissimilarity being the opposite needed in this strategy. We normalise the distance value and subtract from 1, in this way a similarity measure that can operate jointly with the SVM output is obtained. β parameter shown in algorithm 3 represents the weight that receives the similarity measure over the SVM output. In this work β is set up to 0.25

Algorithm 3 shows the procedure of the Density-Weighted method query strategy used in this work. In order to implement this strategy, a similarity measure between all samples needs to be calculated. In this work the Euclidean distance is used as a similarity measure.

5.6 Evaluation measures

In order to measure the previously presented active learning query strategies and to be able to compare with the passive learning methodology, the following evaluation rules have been used. First, we need to present the concepts of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

- TP are all those sentences correctly classified as 1
- TN are all those sentences correctly classified as -1
- FP are all those sentences incorrectly classified as 1
- FN are all those sentences incorrectly classified as -1

Evaluation rules used in this project in order to evaluate the query strategies presented in this work are:

$$Precision = \frac{TP}{TP + FP} \tag{5.6.1}$$

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN} \tag{5.6.2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5.6.3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.6.4}$$

Binary classifiers are usually evaluated with performance measures such as sensitivity and specificity. Alternative measures such as Precision/Recall are used less frequent. In this work results are presented through recall and accuracy because the methodology has been applied to a strong imbalanced data set in which the number of negatives outweighs the number of positives significantly. Recall plots provide the viewer with an accurate prediction of future classification performance due to the fact that they evaluate the fraction of true positives among positive predictions[18].

6 Results

In this section the results of the comparison between AL strategies and versus PL methodology are presented.

6.1 Recall vs Iterations

In this section the results of the recall achieved during the learning process are presented. These results are obtained using a 10-fold Cross-Validation.

Available document language:

Figure 6.1 shows the recall achieved by each of the different AL and the PL methods in *available document language* detection. The PL method needed 18 iterations to achieve the maximum recall, 84.58%, obtained with this database. The AL methods achieved the maximum recall in just 12 iterations. At the 12^{th} iteration, PL reaches only the 72.67% of recall. AL strategies needed 5 iterations to achieve the 80% of recall. In this problem there is not any significant difference between the AL strategies.

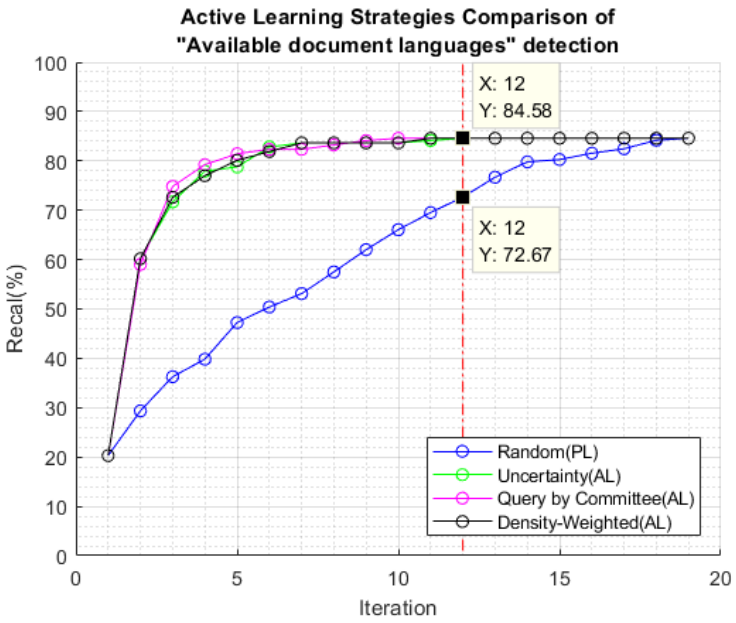


Figure 6.1: Recall of *Available document language* detection.

Exit charge:

Figure 6.2 shows the results of the recall obtained during the learning process in *exit charge* detection. As shown in this figure the difference between AL and PL is significant. All AL strategies only needed 3 iterations to achieve the maximum recall, 92%, and PL needed the enormous quantity of 20 iterations to achieve this recall. In the 3^{rd} iteration, PL reached a recall of 87.87%.

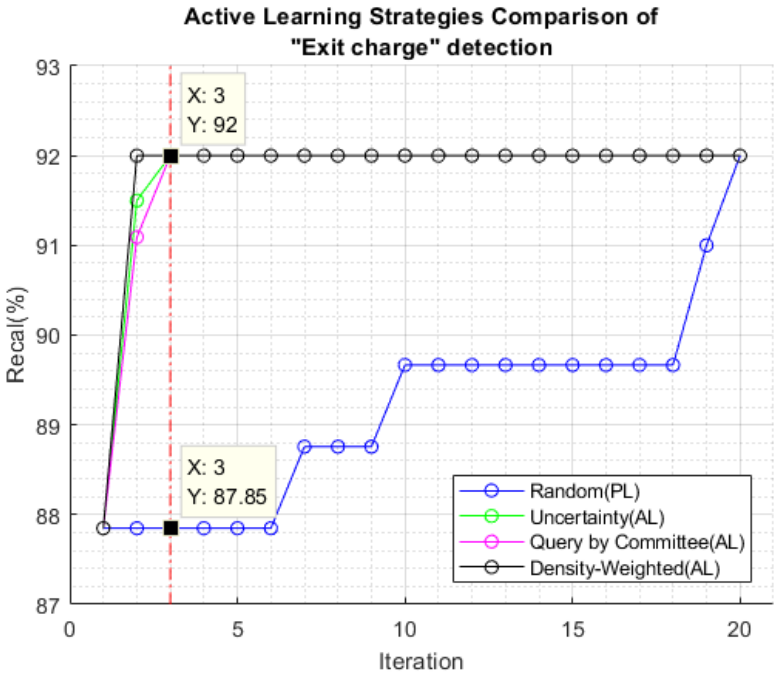


Figure 6.2: *Exit charge* detection recall.

Fund launch date:

The last proposed problem studied in this work is *fund launch date* detection. Figure 6.3 shows the recall achieved by AL and PL. As shown in this figure, AL needed 25 iterations to achieve 89.61% of recall and PL 30. However the major difference can be appreciated in the first 10 iterations were the AL strategies raised the recall more rapid than the PL strategy.

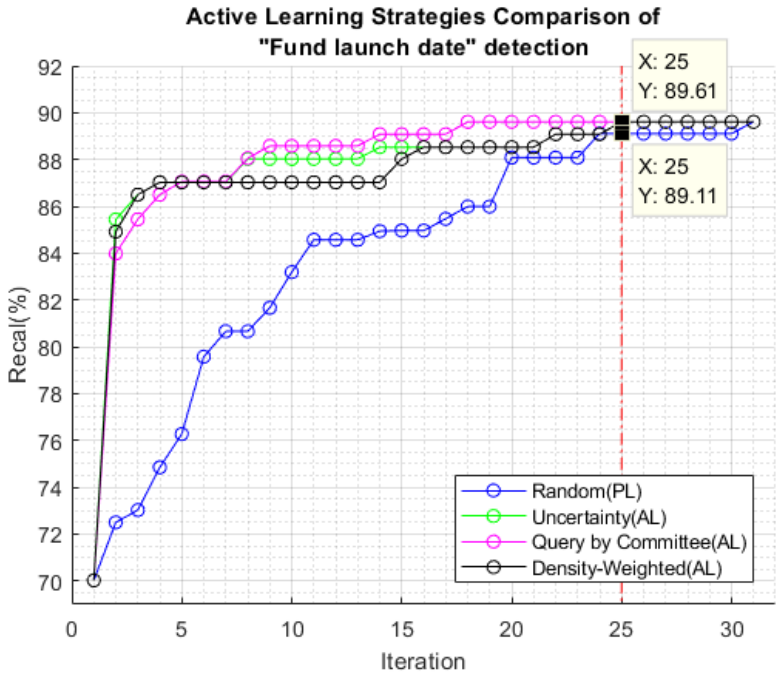


Figure 6.3: *Fund launch date* detection recall.

6.2 Accuracy vs Iterations

In this section the accuracy results achieved during the learning process in the three classification problems proposed are presented.

Available document language:

Figure 6.4 shows the accuracy achieved by each of the different AL strategies and the PL methodology in *available document language* detection. As shown in this figure, PL method needs 18 iterations to achieve the maximum accuracy obtained with this database, 99.8%. Whereas AL strategies were able to achieve the maximum recall in just 12 iterations.

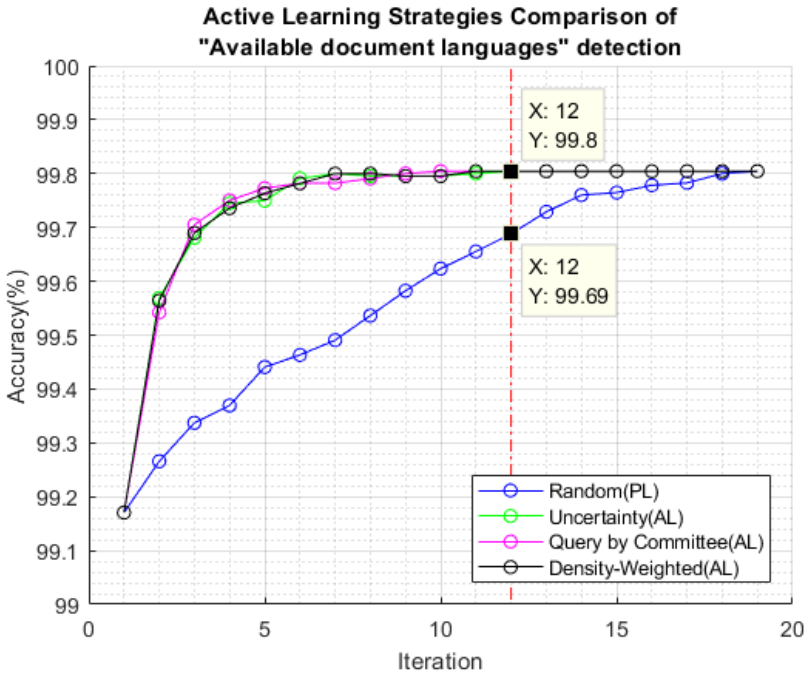


Figure 6.4: *Available document language* detection accuracy.

Exit charge:

Figure 6.5 shows the accuracy obtained during the learning process in *exit charge* detection. As shown in this figure the difference between AL and PL is quite significant. All AL learning strategies need just 3 iterations to achieve the highest accuracy, 99.92% in this problem, however PL needs 20 iterations to achieve the same recall.

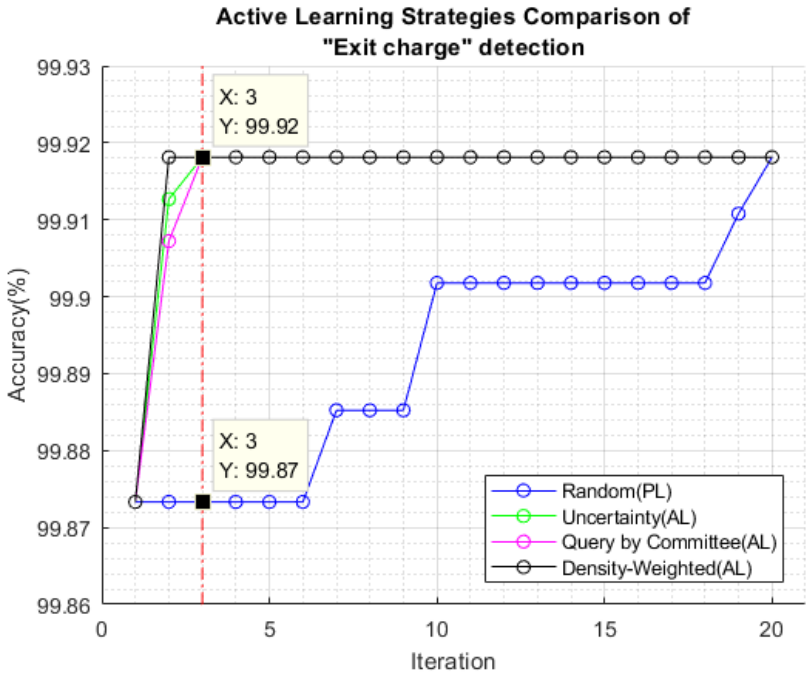


Figure 6.5: *Exit charge* detection accuracy.

Fund launch date:

Accuracy obtained in the last proposed studied problem of *fund launch date* detection is shown in figure 6.6. As shown in this figure, AL needs 25 iterations to achieve the 89.61% of recall. Although, the difference can be appreciated in the first 10 iterations were the AL strategies raise the recall more rapidly than the PL strategy.

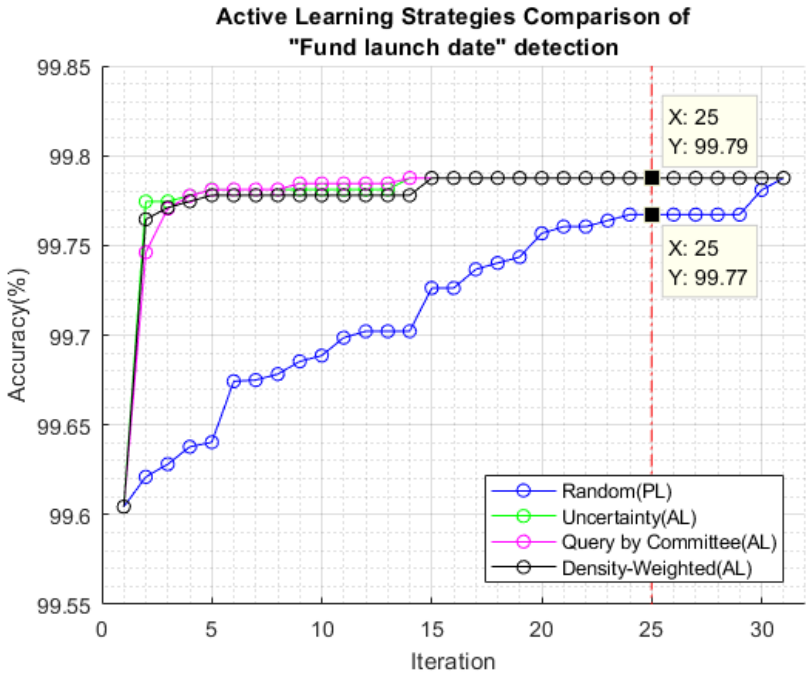


Figure 6.6: *Fund launch date* detection accuracy.

6.3 Time vs Iterations

In this section the results of the time consumption during the learning process by AL and PL methodologies are presented.

Available document language:

Figure 6.7 shows the time consumed by each methodology in *fund launch date detection*. In blue, there is depicted the number of seconds that the PL methodology needs for the training. As it has been shown in previous two sections, AL methodologies achieve the maximum recall in iteration 12. In this figure, it is shown that in the 12^{th} iteration *uncertainty sampling* and *density-weighted* needed almost the same time to compute 12 iterations. On the other hand, *query by committee* needed much more time to compute these 12 iterations. This is because *query by committee* needed to compute 2 different SVM models to query those samples in which the classification disagree more. PL needs less time to compute each iteration because the samples are selected random without computing the probability of classification of any sample.

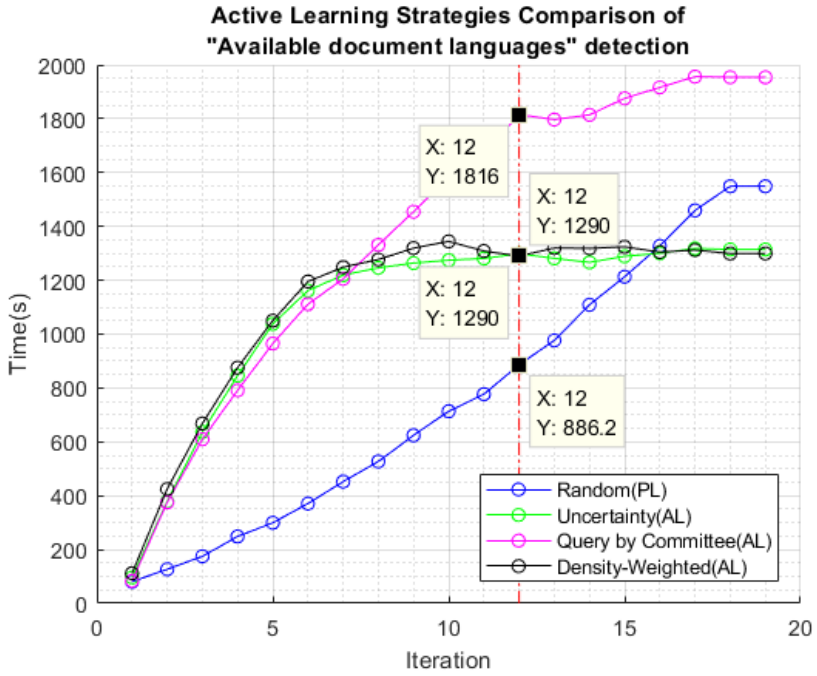


Figure 6.7: *Available document language* time.

Exit charge:

Figure 6.8 shows the time consumption for *exit charge* detection. Figure 6.9 shows a zoom of figure 6.8 at the first 5 iterations. In this figure, it can be seen as well that *query by committee* needed more time together with *density-weighted* method, although the big difference in time consumption can be appreciated in the last iterations in figure 6.8.

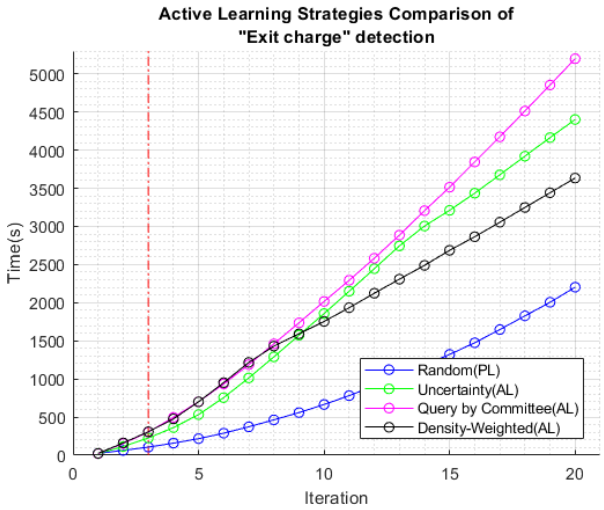


Figure 6.8: *Exit charge* time.

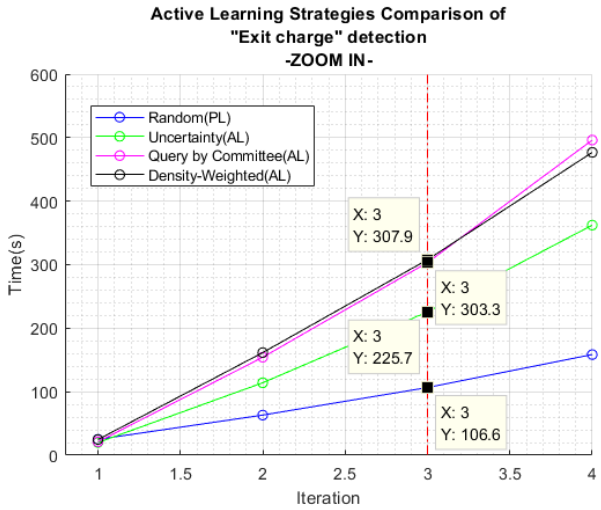


Figure 6.9: *Exit charge* time zoom.

Fund launch date:

Fund launch date is the last proposed studied problem. *Fund launch date* detection is shown in figure 6.10. This figure shows again that *query by committee* needs the double of time than the other two AL strategies and the PL methodology to compute the 25^{th} iteration where the maximum recall and accuracy where achieved.

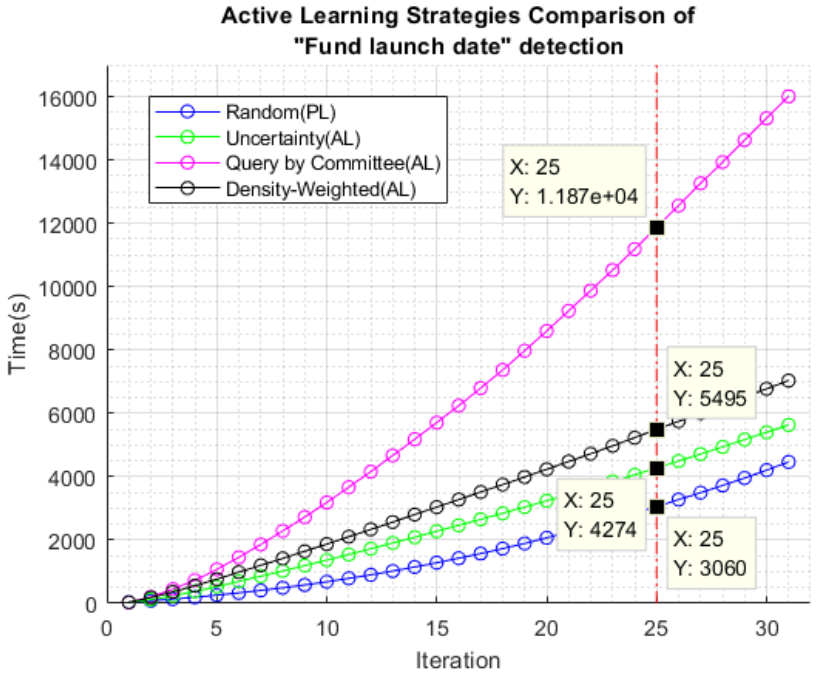


Figure 6.10: *Fund launch date* time.

6.4 Discussion

Results obtained in this work demonstrates that AL strategies allows to obtain a better system than PL methodology. This has been proven through three different text classification problems, *available document languages*, *exit charge* and *fund launch date* detection. The three different classification problems have demonstrated that AL methodology improves the performance of the system faster. In other words, the enterprise costs are substantially reduced by implementing any of the AL strategies proposed in this work.

AL strategies have given similar results in recall and accuracy. Any of the strategies are capable to offer better results than the PL methodology. Meanwhike, time consumption results have helped to discard one of the AL strategies. Time consumption shows that *query by committee* needed more time to compute each learning iteration, to a maximum of twice the time in one classification problem.

Finally, it was concluded that AL methodology helped us to improve the performance of the system by diminishing the time and resources needed. Furthermore, one of the AL strategies was discarded because of the time consumption requires: *query by committee* is discarded. The AL strategies that improved the system in less iterations and needed less time to compute each iteration were *uncertainty sampling* and *density weighted*.

Even although, these last two selected strategies are both capable to achieve similar results. *Density weighted* needed a previous step in order to be implemented, the similarity measure had to be computed. This previous step makes that the best AL strategy implemented in this work is *uncertainty sampling* method.

7 Conclusion and future work

This chapter briefly summarises the thesis, discusses its findings and contributions and outlines directions for future research.

Active Learning is a methodology that can radically accelerate the learning process of many Machine Learning projects and greatly reduce human effort and economic cost. Active Learning prioritises which data is the most confusion about and requests just those labels instead of collecting all labels of the pool of data.

Active Learning methodology has been implemented in a real application that consists of automatically capturing relevant information from Key Investment Information Documents (KIID). Those documents aim to help investors to understand the nature and key risks of the product in order to make a more informed investment decision. The information that has been considered are: *exit charge*, *fund launch date* and *available languages*.

The first phase for obtaining a tool that automatically extracts this information from KIIDs is the learning phase. In this phase a training set of labelled examples must be obtained. These labelled examples are composed of pieces of text from KIIDs. These pieces of text have been obtained semi-automatically with the aid of a specific tool developed by the company where I do my internship. This tool associates to each piece of text 5 different variables or features: *content*, *left context*, *right context*, *bottom context* and *section title*. The label of this piece of text, that is, whether this piece of text is related or not to the desired information (for instance *Exit charge*) is obtained manually. In fact, the positive examples were labelled by hand and the negative examples are obtained automatically with the help of the labelled tool.

The machine learning algorithm used for classification was the well-known Support Vector Machines. The examples were translated into binary vectors through Uni-grams and Bi-grams generating a vocabulary of *one* and *two* words and identifying whether the text feature contains or not these N-grams ($N = 1$ or 2).

Four approaches has been considered to select examples to be labelled: *uncertainty sampling*, *query by committee*, *density weighted* and *random sampling*. The first three of them are Active Learning methods and the last one is a Passive Learning method implemented as reference. 10-fold cross-validation was used to assess each of the approaches implemented and three evaluation measures were applied: recall, accuracy and computational time.

Taking into account both, recall and accuracy, it was concluded that all the Active Learning strategies performed similarly and improved the random sampling. Analysing the time consumption allowed us to discard the *query by committee* approach. Taking into account that the *density-weighted* method needed a previous step in order to be implemented (the similarity measure), the simpler uncertainty sampling method was the approach that was selected for labelling.

Next step will be to add to the semi-automatic labelled tool developed by the company this Active Learning characteristic. Once a large set of KIIDs is supplied, this tool will start extracting a few pieces or text from them and will present to be labelled. From this small set of labelled examples, the tool will inspect the rest of KIIDs in order to select other pieces of text to be labelled using the *uncertainty sampling* method. This process will be carried out sequentially until a certain stop criterion is satisfied. In this way, to obtain an equally efficient model, a smaller number of examples will be needed.

We can extend this problem to other kinds of information besides the three proposed in this work. In addition, similar tool could be applied in other similar problems of extracting information from documents.

Bibliography

[1] F. Internationa, "Investment fund comparison tool 1," Website, 8 2009, last checked: 2010-09-30. [Online]. Available: https://www.fidelity.co.uk/investing/chart-and-compare

[2] Moneywinse, "Investment fund comparison tool 2," Website, 9 2010, last checked: 2010-09-30. [Online]. Available: https://www.moneywise.co.uk/compare-funds-and-investment-trusts

[3] Vanguard, "Investment fund comparison tool 3," Website, 9 2010, last checked: 2010-09-30. [Online]. Available: https://personal.vanguard.com/us/faces/JSP/Funds/Compare/CompareEntryContent.jsp

[4] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *Proceedings of the NIPS workshop on cost-sensitive learning*. Vancouver, Canada, 2008, pp. 1–10.

[5] E. B. Baum and K. Lang, "Query learning can work poorly when a human oracle is used," in *International joint conference on neural networks*, vol. 8, 1992, p. 8.

[6] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203 – 226, 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370282900406

[7] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.

[8] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *International Symposium on Intelligent Data Analysis*. Springer, 2001, pp. 309–318.

[9] C. E. Shannon, "A mathematical theory of communication (parts i and ii)," *Bell System technical journal*, pp. 379–423, 1948.

[10] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 287–294.

[11] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 150–157.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[13] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[14] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[15] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[16] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '10. New York, NY, USA: ACM, 2010, pp. 841–842. [Online]. Available: http://doi.acm.org/10.1145/1835449.1835643

[17] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[18] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, pp. 1–21, 03 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0118432